

Except as otherwise permitted under the copyright, Designs and Patents Act 1988, this thesis may only be produced, stored or transmitted in any form or by any means with the prior permission in writing of the author. The author asserts his/her right to be identified as such in accordance with the terms of the copyright, Designs and Patents Act 1988.



# **Context-aware Personal Learning Environment**

PhD Thesis

Mafawez Alharbi

This thesis is submitted to the De Montfort University UK in partial fulfilment of the requirement for the degree of Doctor of Philosophy(PhD).

Software Technology Research Laboratory (STRL)

De Montfort University

United Kingdom

**January 2014**

Copyright@2014 De Montfort University. All right reserved.

**Declaration**

I declare that this work is done by me and it is original. This work is submitted for the degree of Doctor of Philosophy at Software Technology Research Laboratory (STRL) at De Montfort University (DMU), United Kingdom.

This work has not been submitted for any other degrees or professional qualification. The work was undertaken between January 2011 and January 2014.

Mafawez Alharbi

## **Abstract**

Research is now shifting away from Virtual Learning Environments (VLEs) and towards the use of the Personal Learning Environment (PLE). A review of a number of PLE architectures are presented in the literature, and while they convey well the concept of a PLE, nevertheless they could best be described as high-level architectures, (sometimes referred to as frameworks in the literature), which focus mainly the functionality of PLEs. In particular, there is little published which gives a detailed designed of a PLE architecture. Moreover, the published work focuses largely on the support for lifelong learning and formal / informal learning; these are two of the main limitations of VLEs.

However, this study argues that unexplored potential remains, as there is scope for PLEs to cover more areas. To the best of our knowledge, none of the existing PLE architectures have context-aware systems embedded within their architecture. There is no intelligence in these architectures to filter the e-resources and to predict the user need. In addition, the current PLE architectures are not dynamic; it cannot adopt the user current situation. The user of the current PLE architectures receives too much e-resource.

The architecture proposed in this research incorporates a context-aware engine. Thus there is intelligence built into the architecture and thus the PLE system is automatically responsive to the context information. There are three types of sensors in any context-aware system (physical, virtual and logical), and these are the elements of the system that gather the context information. In this research, the emphasis will be on virtual sensors which gather the information from virtual space; virtual space here includes any systems which produce information as a set of results. Thus, the context-aware architecture and the implementation of the context-aware engine are major contributions of the work.

## **Dedication**

I would like to dedicate this work

*To my mother and father*

*To my wife Shaza and daughters Lean and Reema*

## Acknowledgment

In the name of **ALLAH (God)**. I give praise and thanks to Him for supporting me to complete this thesis.

This thesis could not have been completed without the recommendations and encouragements of many people. Some of these are as follows:

I would like to give a big thank to my first supervisor, **Dr. Amelia Platt**, who provided me with supervision and guidance. It has been a pleasure to have the opportunity to work with you while completing this work. Also for her scientific support, and support in developing my thinking. She gave me the motivation to start this research.

Also, I would like to thank my second supervisor, **Dr. Ali Al-Bayati**, for his suggestions and comments. I had exciting and enjoyable discussions when he was giving me advice. Ali encouraged and challenged me throughout my academic journey. I am grateful for his comments on technical issues involved in the research.

I would like to expand my thanks to my family. My mother and father who pray and support me, they sacrifice their lives for me. Also, I would like to thank my beloved wife Shaza, who has been at my side always, giving me her love and support through this work. My thanks and love go to my beautiful daughters (Lean and Reema) who give me happiness by playing with them and looking into their eyes.

Studying at DMU has been the best time I have ever had in my life. I would like to express my thanks to all the staff and colleagues in STRL Group.

## **List of publications**

[1]- Alharbi, M.T., A. Platt, and A.H. Al-Bayatti. Context-aware Personal Learning Environment. International Conference for Internet Technology and Secured Transactions, 2012 IEEE.

[2]- Alharbi, M.T., A. Platt, and A.H. Al-Bayatti., " Personal Learning Environment", International Journal for e-Learning Security (IJeLS), Volume 4, Issue 1/2, 2013.

# Contents

Declaration.....	i
Abstract .....	ii
Dedication .....	iii
Acknowledgment .....	iv
List of publications.....	v
Contents .....	vi
List of Acronyms .....	xi
List of Figures .....	xii
List of Tables.....	xv
Chapter 1: Introduction .....	1
1.1. Introduction .....	2
1.2. Problem description and Research question.....	3
1.3. Research contribution.....	4
1.4. Research Methodology.....	4
1.5. Criteria for success .....	6
1.6. Research organisation.....	6
Chapter 2: Literature review .....	9
2.1. Virtual Learning Environment .....	10
2.1.1. Introduction.....	10
2.1.2. VLE Definition .....	10



2.1.3. Advantages of VLE .....	10
2.1.4. Disadvantages of VLE .....	11
2.1.5. Summary of major disadvantages of VLE.....	12
2.2. Personal Learning Environment .....	12
2.2.1. PLE Definition .....	12
2.2.2. The Requirements of PLE.....	14
2.2.3. Examples of PLE .....	15
2.2.4. Differences between VLEs and PLEs.....	17
2.2.5. Existing PLE Architectures/Framework and critical review .....	18
2.2.6. PLE Based on Web 2.0 .....	22
2.2.7. Lifelong Learning .....	23
2.3. Context-aware system .....	24
2.3.1. Definition of Context-aware System .....	25
2.3.2. Sensors .....	27
2.3.3. Context-aware Architecture.....	28
2.3.4. Context-aware System Development.....	30
2.3.5. Personalised Service on a Context-aware System .....	32
2.3.6. Example of a Context-aware System.....	33
2.3.7. A summary on how the Context-aware system is important to PLE:.....	33
2.4. Summary .....	34
Chapter 3: Proposed architecture .....	35
3.1. Requirement and motivation .....	36
3.2 Proposal for a novel context-aware Personal Learning Environment architecture	37
3.2.1. Personal Manager (PLE service) .....	38
3.2.2 Context-Aware Engine (PLE service) .....	39

3.3. A walk-through of the architecture.....	47
3.3.1. Stage 1: system registration .....	48
3.3.2. Stage 2: Detection of context.....	49
3.3.3. Stage 3: Generic stage.....	51
3.4. Advantages of the proposed context-aware PLE architecture.....	54
3.5. Limitations of the context-aware PLE architecture.....	54
3.6. Implementation Considerations.....	55
3.7. Case study.....	55
3.7.1. Introduction.....	55
3.7.2. Jenny, PhD student .....	56
3.7.3. Requirements study of Jenny:.....	56
3.7.4. The interface of PLE application .....	56
3.7.5. Scenarios.....	65
First scenario:.....	65
Second scenario: .....	67
3.8. Summary .....	69
Chapter 4: Implementation of the Predication component .....	70
4.1. Artificial Intelligence .....	71
4.2. Bayesian networks .....	73
4.2.1. Static Bayesian network .....	74
4.2.2. Pattern recognition .....	75
4.2.3. BN topology (structure).....	75
4.2.4. Conditional Probability Tables (CPT).....	77
4.2.5. Conditional independence .....	80
4.2.6. Inference in a Bayesian network .....	83

4.2.7. Dynamic Bayesian networks (DBN) .....	85
4.2.8. Bayesian network software.....	86
4.3. Overview of the implementation.....	88
4.4. First DBN model.....	89
4.4.1. Defining the network variables (nodes). ....	89
4.4.2. Drawing the Network graph .....	91
4.4.3. Parameterising the first DBN model .....	92
4.4.4. Inference the first DBN model. ....	95
4.5. Second DBN model .....	96
4.5.1. Defining the network variables (node).....	96
4.5.2. Drawing the network graph.....	98
4.5.3. Parameterising the second DBN model.....	99
4.5.4. Inference the second DBN model.....	101
4.6. Summary .....	101
Chapter 5: Evaluation and Simulation .....	103
5.1. Introduction.....	104
5.2. Evaluation of the proposed architecture.....	104
5.2.1. Available PLE architectures .....	104
5.2.2. Criteria for assessment and critical analysis .....	105
5.4.3 Discussion and result .....	107
5.3. Simulation of the DBN proposed models .....	109
5.3.1. Introduction .....	109
5.3.2. Simulation of the first model.....	110
5.3.3. Simulation of the second model .....	115
5.4. Summary .....	119

Chapter 6: Conclusion and future work .....	120
6.1. Research summary .....	121
6.2. Contributions.....	122
6.3. Success criteria revisited .....	123
6.4. Future work .....	124
References .....	125
Appendix A: Survey .....	137
Appendix B: Transfer data file to BN software.....	140
Appendix C: Data file (results from the survey) .....	144
Appendix D: Creating DBN .....	151

## **List of Acronyms**

VLE: Virtual Learning Environment

PLE: Personal Learning Environment

BN: Bayesian Network

DBN: Dynamic Bayesian Network

API: Application Program Interface

GeNIe: Graphical Network Interface

AI: Artificial Intelligence

JISC: Joint Information System Committee

JSP: Java server pages

VFS: Virtual File Store

ICT: Information Communication Technology

LMS: Learning Management System

APLE: Adaptable Personal Learning Environment

GPS: Global Positioning System

UML: Unified Modelling Language

CCA: Calculus of Context-aware Ambient

NAMA: Need Aware Multi Agent

GUI: Graphical User Interface

## List of Figures

Figure 2. 1: Colloquia .....	16
Figure 2. 2: Future VLE - the visual version .....	18
Figure 2. 3: PLE model .....	19
Figure 2. 4: PLE based on free web services .....	20
Figure 2. 5: Personal Learning Environment Framework (PLEF).....	21
Figure 2. 6: PLE Framework.....	22
Figure 2. 7: Context-aware system functions.....	26
Figure 2. 8: Context broker architecture (CoBrA) .....	29
Figure 2. 9: Infrastructure for context-aware healthcare.....	30
Figure 2. 10: Context-aware System Development .....	31
Figure 2. 11: A general architecture of context-aware system.....	32
Figure 3. 1: Overview of the proposed high-level PLE architecture .....	38
Figure 3. 2: Context-aware Engine subsystems .....	40
Figure 3. 3: Context Reasoner.....	42
Figure 3. 4: Generic.....	44
Figure 3. 5: User Profile.....	46
Figure 3. 6: Registration stage .....	48
Figure 3. 7: Detection of context stage .....	50
Figure 3. 8: Generic stage .....	52
Figure 3. 9: Jenny interacts with a context-aware PLE.....	57
Figure 3. 10: Jenny's Personal Manager .....	58
Figure 3. 11: Personal Information details .....	59
Figure 3. 12: Ways to select providers .....	60
Figure 3. 13: Three different providers .....	60
Figure 3. 14: Predict provider .....	61
Figure 3. 15: Predicate computer science keyword .....	62

Figure 3. 16: Predication result .....	62
Figure 3. 17: Third party .....	63
Figure 3. 18: Feedback.....	64
Figure 4.1: BN structure, step 1 .....	76
Figure 4.2: BN structure, step 2 .....	76
Figure 4.3: BN structure, step 3 .....	77
Figure 4.4: Quality level .....	78
Figure 4. 5: Quantity level .....	79
Figure 4. 6: Causal chains .....	81
Figure 4. 7: Common causes .....	82
Figure 4. 8: Common effects.....	83
Figure 4. 9: One path.....	84
Figure 4. 10: Two paths .....	85
Figure 4. 11: DBN.....	86
Figure 4. 12: GeNIe user interface .....	88
Figure 4. 13: DBN provider model .....	92
Figure 4. 14: Polytrees inference .....	96
Figure 4.15: DBN computer science keyword model.....	98
Figure 5. 1: Summary of comparisons .....	109
Figure 5. 2: Time step count .....	110
Figure 5. 3: Select the states of the node in GeNIe software .....	111
Figure 5. 4: Highest level of provider's state.....	114
Figure 5. 5: lowest level of provider's state .....	114
Figure 5. 6: Highest level of the computer science's state.....	118
Figure 5. 7: lowest level the provider's state .....	118

Figure B.1: Generate a data file in GeNIe, step 1 .....	140
Figure B.2: Generate a data file in GeNIe, step 2 .....	141
Figure B.3: Generate a data file in GeNIe, step 3 .....	142
Figure B.4: Generate a data file in GeNIe, step 4 .....	143
Figure C. 1: Data file for the first model, page 1 .....	144
Figure C. 2: Data file for the first model, page 2 .....	145
Figure C. 3: Data file for the first model, page 3 .....	146
Figure C.4: Data file for the first model, page 4 .....	147
Figure C.5: Data file for the second model, page 1 .....	148
Figure C.6: Data file for the second model, page 2 .....	149
Figure C.7: Data file for the second model, page 3 .....	150
Figure D.1: Creating DBN, step 1 .....	151
Figure D.2: Creating DBN, step 2 .....	152
Figure D.3: Creating DBN, step 3 .....	153
Figure D.4: Creating DBN, step 4 A .....	154
Figure D.5: Creating DBN, step 4 B .....	155
Figure D.6: Creating DBN, step 5 A .....	156
Figure D.7: Creating DBN, step 5 B .....	157
Figure D.8: Creating DBN, step 6 .....	158



## List of Tables

Table 2. 1: Major Differences between VLEs and PLEs .....	18
Table 2. 2: Physical sensors .....	28
Table 3.1: Result step 2 .....	65
Table 3.2: Result step 3 .....	66
Table 3.3: Result step 4 .....	66
Table 3.4: Result step 5 .....	66
Table 3.5: Result step 6 .....	67
Table 4. 1: CPT for node A .....	79
Table 4. 2: The possible states of the variables .....	91
Table 4. 3: The states of the hypothesis node .....	91
Table 4. 4: CPT for provider node .....	93
Table 4. 5: CPT for type of material node .....	94
Table 4. 6: CPT for membership node .....	94
Table 4. 7: CPT for other nodes .....	95
Table 4. 8 The states of the variables .....	97
Table 4. 9: The states of the hypothesis node .....	98
Table 4. 10: CPT for artificial intelligence node .....	99
Table 4. 11: CPT for computer security node .....	99
Table 4. 12: CPT for software engineering node .....	100
Table 4. 13: CPT for the hypothesis node .....	100
Table 4. 14: CPT for other nodes .....	101
Table 5. 1: General information on existing PLE architectures .....	104

Table 5. 2: Summary of comparisons .....	108
Table 5. 3: Combinations of the states of the input nodes .....	112
Table 5. 4: Inference for first model .....	113
Table 5. 5: combinations of the states of the input nodes .....	116
Table 5. 6: Inference for second model.....	117

## Chapter 1: Introduction

### **Chapter 1: Introduction**

#### **Objectives:**

- To introduce the study
- To describe the problem and clarify the research question
- To highlight the original contributions
- To describe the research methodology
- To present criteria for success
- To explain the organisation of the thesis

## Chapter 1: Introduction

### **1.1. Introduction**

Virtual Learning Environments (VLEs) have become popular in higher education in recent years, due to their ability to provide additional and flexible solutions for students and researchers. However, the limitations that affect VLEs have also led to the development of a new generation of VLE – the Personal Learning Environment (PLE). PLEs avoid the limitations of VLEs by incorporating novel features, such as allowing students to control and utilise new type of applications, including Web 2.0 and social networks. Whilst PLEs were primarily designed to offer a resolution to some of the drawbacks of VLEs, arguably their potential is much greater. This study will show that PLEs are not yet well defined, and that they too have some limitations. This research argues that PLEs can support more than mere learning in education; they can also support users in any aspect of their lives that requires them to find and utilise information available on-line. In addition, it argues that a context-aware system can support the construction of PLE architecture.

Weiser first introduced ubiquitous computing (pervasive systems) in 1991 [108]. Context-aware systems are a type of pervasive system, and they are viewed by computer scientists as a mature technology. Dey and Abowd in [38] defined a context-aware system as, “a system that uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task”. Context-aware systems are able to gather contextual information from a variety of sources, without explicit user interaction, and to adapt their operations accordingly. Context-aware systems can also be easily integrated within any service domain, such as healthcare, commerce, learning and transport.

The architecture proposed in this research incorporates a context-aware engine. Thus, intelligence is built into the architecture, and that intelligence is automatically responsive to the contextual information. There are three types of sensors in any context-aware system: physical (e.g. camera), virtual (e.g. operating system) and logical (e.g. using camera and operating system); and these are the elements of the system that gather the contextual information [42]. In this research, the emphasis is on virtual sensors, which gather information from virtual space; virtual space here includes any system that produces information as a set of results.

## Chapter 1: Introduction

### **1.2. Problem description and Research question**

Research is now shifting away from Virtual Learning Environments (VLEs) and towards the use of the Personal Learning Environment (PLE). A review of a number of PLE architectures are presented in the literature, and while they convey well the concept of a PLE, nevertheless they could best be described as high-level architectures, (sometimes referred to as frameworks in the literature), which focus mainly the functionality of PLEs. In particular, there is little published which gives a detailed designed of a PLE architecture. Moreover, the published work focuses largely on the support for lifelong learning and formal / informal learning; these are two of the main limitations of VLEs.

However, this study argues that unexplored potential remains in PLE system, as there is scope for PLEs to cover more areas. To the best of our knowledge, none of the existing PLE architectures have context-aware systems completely embedded within their architecture. Context-awareness is a vital aspect of any intelligent system, as its input changes. There is no intelligence in the current PLE architectures to predict what e-resources the user may need. In addition, the user receives many e-resources when using the existing architectures. The reasoning part is missing from all other PLE architectures. The definition of the word context in this research is “any information that describes the behaviour of the user; for example, the year of study for research student.

The overall research question is as follows

***Can a context-aware Personal Learning Environment architecture be built?***

The major goal of this research is to address the above question. In order to answer the research question, three sub-questions are presented as follows:

- How can the context-aware system be integrated into Personal Learning Environment architecture?
- How can the newly proposed architecture be correctly implemented using a suitable technique?
- How can the proposed architecture be evaluated?

## Chapter 1: Introduction

### **1.3. Research contribution**

This research presents the following contributions to the body of knowledge:

- An approach known as “Context-Aware Personal Learning Environment Architecture” is developed. The PLE architecture is automatically responsive to each individual user on demand, and it is able to predict user preferences or interests. The automated refinement of user requests for e-resources enables results to improve over time. The architecture consists of two layers: the top layer is the PLE service and the PLE interface for the user and the bottom layer shows other independent tools or service providers. This contribution is illustrated in Chapter 3, where comprehensive definitions of its components are presented.
- The implementation of the Prediction component in the proposed approach is presented using Bayesian Network (BN) techniques. Two Dynamic BN (DBN) models are presented in the implementation. The first DBN model is used to predict a suitable provider for the user. The second model presented uses keywords from the Computer Science vocabulary to predict a suitable subject for the user. This contribution is presented in Chapter 5, and the simulation for the implementation is presented as part of Chapter 6.

These contributions have been reported at IEEE conference and published in e-learning journal.

### **1.4. Research Methodology**

The methodology used in this research utilises a scientific research technique (constructive research). Constructive research refers to contributing to the body of knowledge in terms of developing, for example, a new architecture, model or algorithm [59]. The proposed approach comprises five phases. The first is the research

## Chapter 1: Introduction

background, and the second focuses on the architecture. The third phase presents suitable artificial intelligence techniques for the proposal architecture. The fourth demonstrates the implementation of the proposed architecture. The final phase concentrates on the evaluation of the proposed architecture and simulation of the DBN models.

All five phases are presented in detail as follows.

- **Phase 1:** Research background and critical review

An introduction and critical review of related work is conducted. Initially, a widespread study of VLEs, PLEs and context-aware systems is undertaken in order to determine a starting point for this research. In addition, this will identify the limitations of existing PLE architectures and the suitability of context-awareness for PLE architecture.

- **Phase 2:** Architecture

This phase identifies the research problem, and then constructs a solution by developing an architecture that is relevant to the context-aware PLE architecture. In addition, this phase describes our contribution to the field.

- **Phase 3:** Artificial intelligence techniques

This part focuses on exploring the available artificial intelligence techniques. The chosen technique will be used to implement the proposed architecture. To model a context-aware system when the context is ‘uncertain’, an AI technique must be used. Thus, the selected technique should be an appropriate one for implementing the proposed approach.

- **Phase 4:** Implementation of the proposed architecture

This phase implements the proposed architecture, which requires the support of the AI component. As an AI component is embedded within the PLE architecture. The implementation of the proposed architecture must focus on the implementation of the AI component because it represents a major challenge to

## **Chapter 1: Introduction**

the functioning of the proposed architecture. The description of the implementation will be based on a case study.

- **Phase 5:** Evaluation and simulation

The evaluation step in this study involves a comparative study with other PLE architectures, in order to demonstrate the strengths of the proposed architecture. The simulation in this research is that to checks that the proposed models satisfy their intended purposes.

### **1.5. Criteria for success**

The criteria by which to determine the success of the thesis will be satisfied by the following:

- The research questions in Section 1.2 must be met.
- The research successfully shows how the proposed architecture differs from others.
- An analysis of why a Bayesian Network was chosen from among the possible reasoning techniques must be performed.
- To implement the proposed architecture, artificial intelligence techniques were used.

### **1.6. Research organisation**

The remainder of this research is organised as follows.

## **Chapter 2: Literature review**

Chapter 2 provides an overview of the three main topics, which are VLEs, PLEs and context-aware systems. The chapter begins by discussing VLEs, offering a definition and discussing their advantages and disadvantages. The chapter then introduces the concept of a PLE, which includes a definition, requirements and examples, and also covers the differences between VLEs and PLE architectures based on Web 2.0. In addition, the major limitations of the existing PLE architectures are presented. Finally, the chapter discusses context-awareness, defining context-aware systems, sensors and



## **Chapter 1: Introduction**

context-aware architecture, describing context-aware system development, presenting an example of a context-aware system, and assessing the suitability of a context-aware system for PLE architecture.

## **Chapter 3: Proposed architecture**

This chapter proposes a novel architecture, which is referred to as a context-aware PLE architecture. It begins by disclosing the requirements and the vision for the PLE. Following this, the chapter presents the proposed architecture by detailing each component and its functions, and showing how these components interact. The chapter then presents a walk-through of the proposed architecture. In addition, the advantages and limitations of the proposed architecture are detailed. After that, implementation considerations are presented. Finally, this chapter ends by providing a case study for the proposed architecture, which demonstrates that the proposed architecture is valid and suitable for use.

## **Chapter 4: Implementation of the Prediction component**

The first part in this chapter illustrates the technique selected, which is a Bayesian Network (BN). This technique is accordingly used in this research to implement the proposed architecture. This part covers the fundamental principles of BN as well as the software used for this technique.

The second part of the chapter considers implementing the Prediction component within the Generic subsystem. The Prediction component has the ability to predict contextual information. The aforementioned AI technique, i.e. a Dynamic Bayesian Network (DBN), is used to implement the Prediction component in the proposed architecture. Two DBN models are presented in this chapter; the first is used to predict a suitable provider for the user, and the second uses keywords from the Computer Science vocabulary to predict a suitable subject for the user.

## **Chapter 5: Evaluation and simulation**

This chapter discusses the evaluation of the proposed architecture and the simulation of the Dynamic Bayesian Network models. Firstly, the chapter discusses the evaluation

## Chapter 1: Introduction

step through a comparison of the proposed approach with other PLE architectures, in order to demonstrate the strengths of the proposed architecture. Secondly, this chapter presents the simulation in this research is that to checks that the proposed models satisfy their intended purposes.

## **Chapter 6: Conclusion and future work**

This chapter discusses the research conclusions. It includes a research summary, identifies original contributions and makes suggestions for future work.

## **Chapter 2: Literature review**

### **Objectives:**

- To present virtual learning environment
- To present personal learning environment
- To describe context-aware systems

## Chapter 2: Literature review

### **2.1. Virtual Learning Environment**

#### **2.1.1. Introduction**

Technology has been used to deliver education and training for a number of decades. However, it could be argued that the use of Virtual Learning Environments (VLEs) in the last decade have brought about the greatest changes in the delivery of education allowing learners to access multimedia course material at any time, from anywhere to suit their individual needs. Thus learners need not attend traditional, face to face, activities in the classroom. Instead, the teaching and learning are done in a virtual space. VLEs are now pervasive at all levels of education.

#### **2.1.2. VLE Definition**

Many definitions have been proposed for VLEs as presented in [1, 2]. Note that VLEs are also known as a Learning Management System (LMS); in this thesis only the term VLE is used. Many of the definitions given are similar, and typically focus on a particular characteristic of VLEs. For the purpose of this study, the definition that will be used is that presented in [2], which defines a VLE as a software system used to deliver online education, taking advantage of web-based methods and tools such as discussion forums, chats and automated tests [3,56]. This definition has been selected because it is appropriate for this research. The further advantages of VLEs will be presented in the next section.

#### **2.1.3. Advantages of VLE**

This section will introduce the advantages of VLE in [4, 5, 6, 56] as outlined below:

- There are increasing numbers of students in higher education around the world, so a VLE can help institutions to reduce the practical pressures of finding enough physical space and resources for them.
- Students and learners may study course modules at anytime and anywhere, unlike in traditional learning environments. Some learners prefer not to have a fixed time or place for learning, so VLEs can help them in this way.

## Chapter 2: Literature review

- Learning by VLE is more economic than traditional learning. Students can study from home without going every day to a university and spending money on transport, etc.
- VLEs can provide access to education for those who have experienced barriers to it in the past, such as students who have special needs or a disability. In addition, learners who have family commitments or individuals with financial difficulties are also able to participate in the learning environment through use of a VLE.
- Educators can use VLEs to manage courses, for example by sending announcements to students, issuing assignments and uploading course material. VLEs can offer student services such as discussion boards, document sharing and off-line lectures or notes. VLEs can thus connect the course participants so as to achieve both effective communication and collaboration. Via VLEs, students and lecturers can enjoy the convenience of using the online delivery of materials and resources.
- VLEs make education available to the broader population. Education is not restricted to only those students who are going to university, but is open to anyone interesting in being educated via a VLE.

With all these advantages of VLEs however, there are also some limitations, as addressed in the next section.

### **2.1.4. Disadvantages of VLE**

Even though VLEs offer many advantages, they also have some limitations, which are introduced in [7, 8 and 9] as reviewed below.

- The most serious disadvantage of VLEs are that on courses and modules students are allocated a restricted time period during which the online course / module materials and tools are available to them; typically this spans the duration of being registered on the module. Thus, once a student has completed the module they no longer have access to the module resources. This means that they cannot go back and review material covered on an earlier module.

## Chapter 2: Literature review

- One of the drawbacks of the VLE is that educators are only responsible for setting the goals of the courses, without taking into account the students' individual requirements. VLEs are not flexible and cannot address the specific needs of the learner.
- Another problem with VLEs is that the direction of information is only one way, from educators to learners. Teachers only have the authority to post material / resources on the VLE.

### **2.1.5. Summary of major disadvantages of VLE**

- VLEs are not flexible and cannot address the specific needs of the learner.
- The courses and modules are available to students for a restricted time period.
- Educators are only responsible for setting the goals of the courses; they cannot take into account the students' individual requirements.

Because of the limitations addressed above, researchers are seeking a new generation of VLEs that will improve upon the current shortcomings. Therefore, research is now shifting away from VLEs and towards the use of the Personal Learning Environment (PLE), which will be discussed in the next section.

## **2.2. Personal Learning Environment**

As noted above learners have access to material relating to a given module for a restricted time period, thus VLEs do not support lifelong learning and other requirements that users might have. Recognising the limitations of VLEs, researchers have to consider how to cover them and, today, there is a marked shift away from VLEs to PLEs [10].

### **2.2.1. PLE Definition**

The PLE is an important area in the field of e-learning, and recent developments in PLEs have heightened the need for a formal definition of what exactly a PLE is; some proposals are reviewed below:

The Joint Information System Committee (JISC) [11] defined a PLE as one that exchanged some or all of the VLE tools for personal tools, including web 2.0 type tools, which integrate with the student's own computer system environment. The tools and

## Chapter 2: Literature review

interface are designed by the learners rather than the institution, as is the case with a VLE. The PLE has also been defined by [8] as being “to provide the learner with a wide variety of different services and hand over control to the learner to select, use, and mash up the services the way one deems fit”.

PLEs are defined by [12]: “PLE is not an application, but rather, a description of the process of learning from a variety of courses and according to one’s personal, context-situated need”. Thus one of the major objectives of a PLE is to widen access to education systems via technology to a student who wishes to manage their own learning. A PLE should support a learner in four respects: to create and manage their learning, communicate with other learners and also integrate formal and informal learning.

The PLE is defined in [13] as a specific virtual space designed by the user with their individualised information, learning, social and recreational needs. A PLE thus takes advantage of new technology and tools. In particular, the PLE is defined as:

- Personal: referring to the interests and the needs of each learner, as for example, children, adults and elderly people will differ in these respects and the tools should therefore be selected by the user.
- Learning: including formal and informal learning.
- Environment: ensuring that everyone has access to the best resources and that a culture of learning is created.

There are seven important points concerning PLEs, which are presented in [14]. These are: what is it, who is using it, how does it work, why is it significant, what are the downsides, where is it going and what are the implications for teaching and learning?

According to Attwell (2007) [15], most researchers agree that a PLE is not an application but is best described as a new approach to using technological innovations for learning. The study argues that if learners control their own learning, then the teacher has as-yet an undefined role in the situation.

While a variety of definitions for a PLE have been suggested, the definition used in this study is that “PLE is the system that has the ability to filter e-resources and predict user

## Chapter 2: Literature review

preferences based on user context”. Note, this covers the use of a PLE for many purposes in addition to traditional formal (institution-based) and informal (private, non-institution-based) academic learning. For instance, the PLE could be used for social or business purposes. A business man can use a PLE system to find recent offers made between businesses, so that he can weigh up his options and evaluate the best contract to pursue. Whereas the user of Facebook might state in his personal preferences that he/she would only like to see pictures of family and friends on their news feed.

### **2.2.2. The Requirements of PLE**

Some studies in the field of the PLE have detailed the requirements; these are reviewed in this section. Downes [16] identifies three aspects of the PLE, which are to satisfy the user’s requirements regarding interaction, usability and relevance. Interaction in a PLE requires users be able to communicate with other users who are also interested in the same subject. During the communication the user can collect additional information; a good example of this is Facebook. Usability supports freedom, in that a user of a PLE should have a choice in selecting specialist tools. Relevance refers to importance of information, as different users might have different specialised interests that are not necessarily shared by others.

In [17], it is argued that a PLE is a new phenomenon in learning, which supports three features: lifelong learning, e-portfolios and control. Lifelong learning means that a learner can use a PLE system that is integrated with different institution’s systems, which requires the learner to use different systems across institutions [18]. PLEs should support e-portfolios to deliver user needs too, and allow the user to store any resources they want. Control in a PLE needs to come via a learner in order to update their PLE.

Harmelen presented a set of dimensions that characterise the space of the PLE [19], reviewed below:

1. Collaborative: a user may collaborate in teaching and learning activities; Colloquia is a good example for collaboration which is a popular VLE application.



## Chapter 2: Literature review

2. Openness: an open system can be extended easily. This dimension effects personalisation, where a user has the ability to capture and use a URL in order to obtain extra facilities.
3. Control: there is a major practical limitation in VLEs, as institutions or education systems have control of the resources. It is important for a new PLE to approach control as being shared between both learner and teacher. As a result, ownership of the PLE is shifted from the institution to the learner and teacher.
4. Institution connectivity: current VLEs support short-term learning activities, whereas PLEs support long-term activities where a learner can access to information any time. A PLE requires connection to multiple institutions and continuing professional development through education. In such cases, a PLE will interoperate with a variety of services.
5. Online/offline: it is important to support both online and offline learning. A good example is the Manchester Framework [21], which can be used as either a VLE or a PLE and the user is able to use the system offline.
6. PLE platform: major platforms such as desktops and laptops are not suitable for a PLE. The use of smaller platform for the hosting of a PLE is considered better, such as being run on a PDA or iPhone.

Some positive features of the PLE framework are presented in [12]. They include: personalised, flexibility and usability. A personalised PLE framework should support a learner, including and using tools, in order to help him/her to create their own PLE that is adapted to the learner's individual situation. With flexibility, a PLE framework gives the learner the ability to integrate, aggregate and combine different learning services depending on their interests. Usability allows a learner to copy and paste elements to personalise and control his/her PLE with minimum effort.

### **2.2.3. Examples of PLE**

Research into PLEs is still at an early stage, consequently there are still very few usable PLE systems available and these are experimental systems only. The Colloquia, Manchester PLE and Portland Personal Learning Environment are examples of such systems and are described below:

## Chapter 2: Literature review

Colloquia is funded by the Joint Information System Committee (JISC), presented in [20] and produced by the TOOMOL project. One of the features is presented in Figure 2.1. Colloquia uses peer-to-peer communication supported by an email server, and maintains information concerning resources, students and activities; both teachers and students can participate in the project. The two main ideas behind Colloquia were to move students from traditional classrooms to group conversations based on learning and to support learners with mobile and personal tools, enabling them to work offline.

Colloquia allows learners to control their personal information and to maintain their learning resources. Each student in Colloquia has permission to access other students' information and the history of the student's last conversation. Teachers have the ability to set up activities and sub-activities. In addition, teacher and learners can add and delete resources at any time. One major drawback of this approach is that the application can only support learning; however it does not support other aspects of life, i.e. business aspects and social aspects.

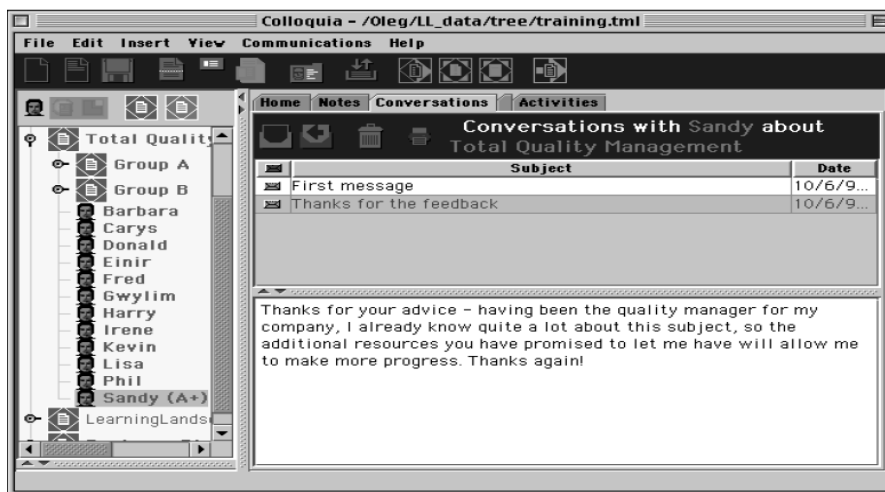


Figure 2. 1: Colloquia [20]

The Manchester PLE includes servlet containers such as users, friendship, groups, activity feeds, tags, tag search, other search facilities [21], underpinned by a Virtual File Store (VFS). It provides servlet and JSP (Java Server Pages) authors with a variety of primitive operations that support the implementation of an e-learning system. The application can be used as either a VLE or a PLE and the user is able to use the system offline, in this case a PLE system. If the system is online, the system will be used as

## Chapter 2: Literature review

either a VLE or a PLE. The main weakness of the Manchester PLE is that it does not use the information in a context-aware way. It is not able to predict users' interests.

The aim of the Portland PLE is to develop a prototype PLE application for ICT-based software and curriculum content in order to meet the needs of adult learners with disabilities [22]. The project was led by Portland College in Nottinghamshire, UK, and the University of Teesside is responsible for the analysis of user requirements. The project identifies the needs of a group of learners with disabilities. The average user of Portland was a young adult learner, aged 16 years old, but each learner has unique needs in term of accessing information. The project supports online communication and learning. The Portland interface is based on a learner profile that was established during the assessment. The project was designed to meet the individual needs of the target user; Portland support learners with low literacy levels are supported by symbol support text and speech output.

In [23], Portland PLE has been extended to adapt to meet the needs of any learner, not just in terms of disabled people. An Adaptable Personal Learning Environment (APLE) takes the concept of Portland in creating an accessible, personalised and flexible learning environment for learner, but the adaptability depends on the recognition of the learner with sensory and cognitive functions. A learner is therefore able to establish a profile with preferences regarding his/her needs. The weakness in this application is that APLE is not designed with the ability to automatically adapt to the current user context; it does not take into account the user's context. If a student is in the third year of their work, the system will not be able to identify that the student is looking for a thesis.

### **2.2.4. Differences between VLEs and PLEs**

There are a number of differences between VLEs and PLEs that have identified in [8]. Others has been aggregated from [15] and presented in Table 2.1 below.

<b>VLEs</b>	<b>PLEs</b>
Institutions only can create and make information available	Users can also create and make information available
Formal learning material only	Includes formal and informal material
Materials are available temporarily for a fixed period	Materials are available indefinitely

## Chapter 2: Literature review

Institutions only have control of the environment	Users also have control of the environment
From teacher to learner	From learner to learner or teacher
Information direction is one-way, from teacher to learner	Information direction is two-way
Content is shared	Content is shared and can also be uploaded

Table 2. 1: Major Differences between VLEs and PLEs

### 2.2.5. Existing PLE Architectures/Framework and critical review

The concept of a PLE is a recent development; consequently there are only a small number of studies considering the possible software to support a PLE. This section will present existing PLE architectures. In some cases the term framework is used instead of architecture, typically when a high level architecture is given.

In 2005, the philosophy of a PLE began to emerge from conversations with different groups of educational technologists. The first PLE architecture was presented by Wilson [24] in 2005. This could be described as a high level architecture and Wilson referred to as the “Future VLE”. The architecture, shown in Figure 2.2, introduces a design pattern with the emphasis on a shift away from the isolated experience of the modular VLE. Perhaps the most serious disadvantage of this particular architecture is that it is reactive, in contrast to proactive, in that there is no intelligence in the architecture to predict what information / resources the user may need in the future, based on what they have done so far.

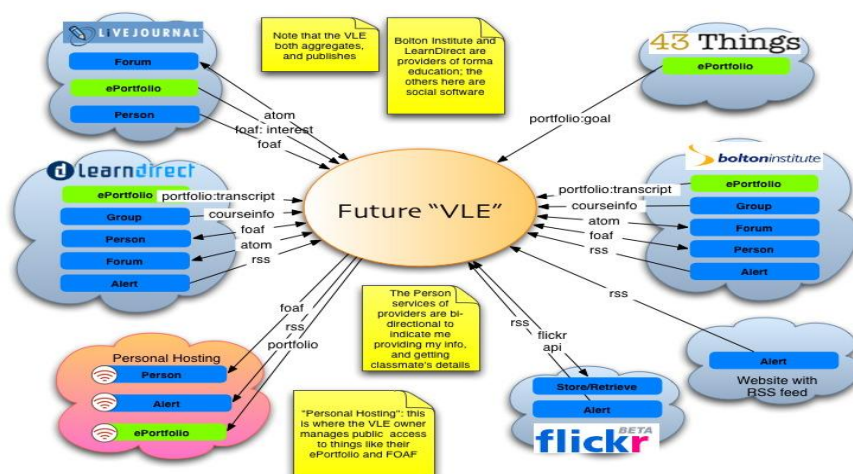


Figure 2. 2: Future VLE - the visual version [24]

## Chapter 2: Literature review

An updated version of Wilson's vision is presented in [25] and shown in Figure 2.3. This updated version includes references to formal and informal learning, work and leisure. It could be argued that this proposal has clarified the basic functionality of the PLE. For example when a user needs to deal with photos, a suitable application (Flickr) is used. A serious weakness of this proposal, however, is that it does not take into account the user context. Furthermore, it is not a dynamic system. This means that it does not use past history to predict future events.

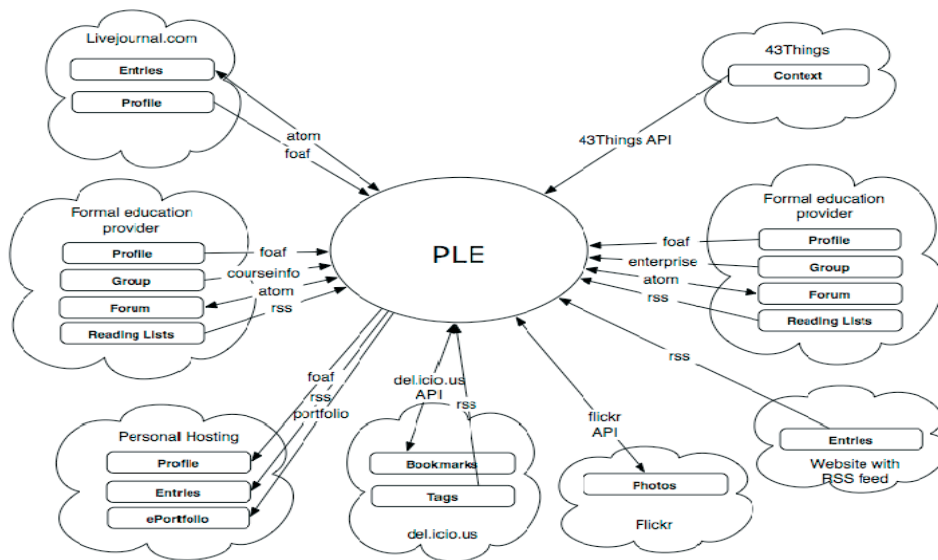


Figure 2. 3: PLE model [25]

Another proposal is shown in Figure 2.4 and presented in [9]. In this model (the term used in place of architecture), the PLE is based on standard web services and includes a web application that allows the user to manage his/her learning using distributed sources. Users interact with an application that is connected to external web services, and the model contains entities involving both clients and the server. The server

## Chapter 2: Literature review

includes administration services, which are responsible for managing all data stored in their PLEs. The system uses Open ID, which means that the user does not have to create another user name, but can use the same user name to access different distributed services. The primary goal of the model is to allow learners to manage their learning, but also to communicate and cooperate with a learning community. However, similar to the previous architectures, a serious weakness is that there is no built in intelligence. This means that the model is not able to provide relevant e-resources, which is a very important requirement for the user.

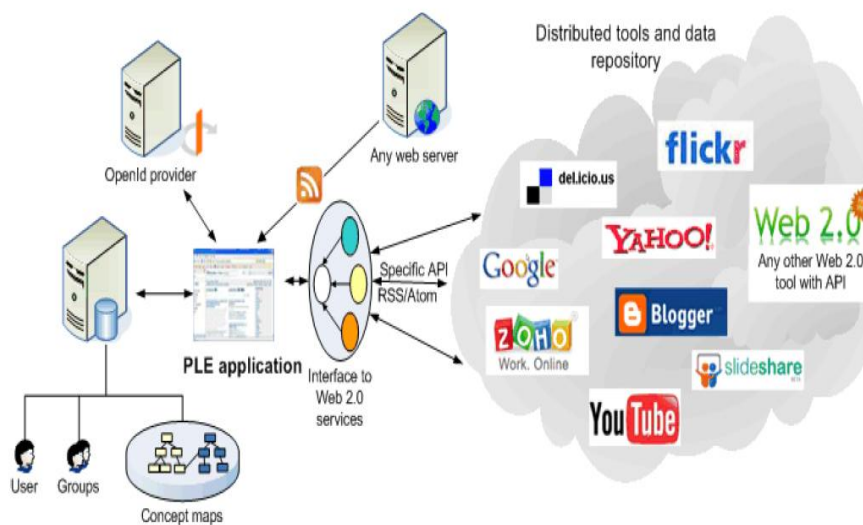


Figure 2. 4: PLE based on free web services [9]

In [26], the author proposed framework for PLEs that supported personalised, informal learning with lifelong-style network; this is shown in Figure 2.5. The framework aims to help learners create their own learning, consisting of different learning services such as media, feeds and widgets. Learners have ability to manage (add, edit and delete), search and review their learning, and are also able to visit other PLEs and give their opinions. The architecture explicitly addresses informal learning, lifelong learning and personalisation. The PLE is based on the Web 2.0 applications. One major drawback of this approach is that the user of the system is likely to receive excessive e-resources from the various applications, much of which will not be relevant.

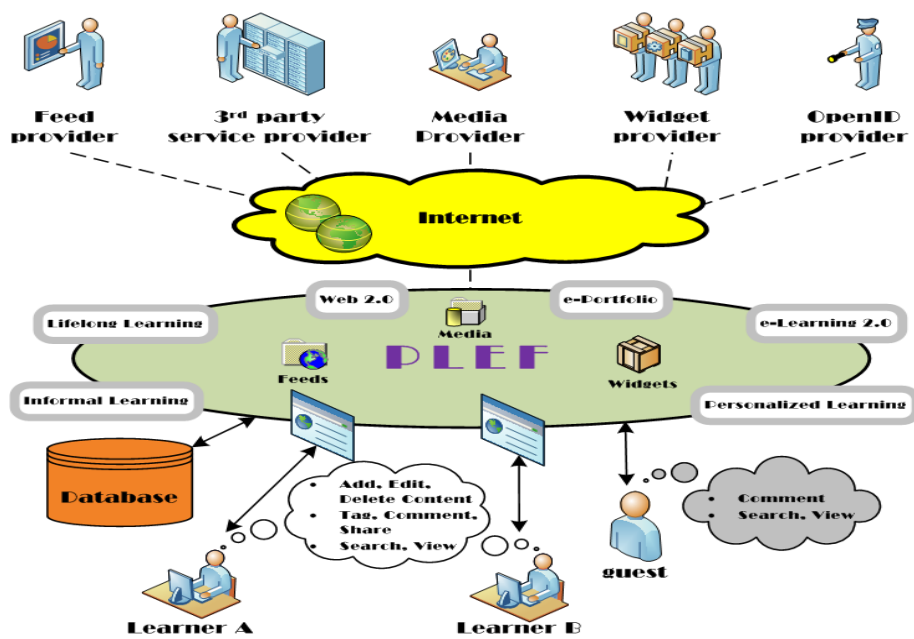


Figure 2. 5: Personal Learning Environment Framework (PLEF) [26]

The final PLE architecture discussed here is presented in [8] and shown in Figure 2.6. The architecture allows learners to have control over their learning, by aggregating, managing, tagging, commenting and sharing their favourite resources. For example feeds, widgets and different media within a personalised space. The architecture supports five important functions:

- Open ID for authentication.
- Commenting and sharing of all PLE components.
- Access control is defined on a PLE page.

## Chapter 2: Literature review

- A tag view of all PLE elements. Learners are able to add tags in order to be able to classify, categorise, search and re-find their PLE elements at a later time.
- Full-text and tag searches of PLE elements.

The major limitation of this proposal is that it is little more than a concept at this stage as it provides no information on how it could be implemented.

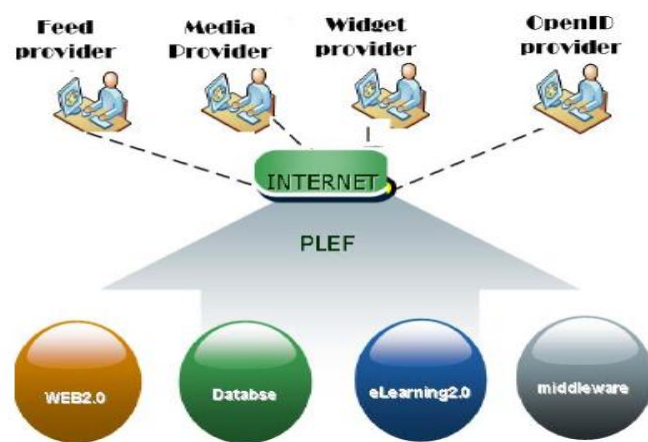


Figure 2. 6: PLE Framework [8]

This section reviews the major limitations of the proposed PLE architecture; these are listed below:

- It does not take into account the user context.
- The user of the system receives excessive e-resources that might not be required by the user.
- It does not provide sufficient detail about how it could be implemented.
- It cannot predict user preferences.
- It does not apply artificial intelligence techniques.

### **2.2.6. PLE Based on Web 2.0**

It has been noted that Web 2.0 technologies provide a good platform for implementing PLEs [27]. Web 2.0 is an umbrella term that includes several new types of web



## Chapter 2: Literature review

applications. The concept of Web 2.0 began in a conference in 2004, when O'Reilly introduced Web 2.0 for the first time. The main feature of Web 2.0 applications is that a user can communicate with his/her peers, interactively, using multi-media. Moreover, Web 2.0 technologies typically provide an Application Program Interface (API), in order to make it easy for developers to build applications which provide the two ways communication needed for the collaboration nature of Web 2.0 applications. This is to make sure that two routes of direction are taken as an action between users [7]. A number of experimental PLEs make use of the features offered by Web 2.0 applications.

### **2.2.7. Lifelong Learning**

Lifelong learning means that learning should be available to learners throughout their lives. Lifelong learning refers to continuously allowing learners to update their knowledge, so they can meet the changing needs in the world. Where knowledge is no longer limited to the classroom and not restricted to the school environment, learning resources are for the community [57]. Thus, the world is described by continuous learning. It demands the continuous building of skills and knowledge throughout life [58].

One of the major features of PLEs, compared to VLEs, is that their resources can be gathered and retained indefinitely. The resources collected by the learners can be available to them until they decide no longer require them. Attwell [15] suggests that PLEs are tools to support lifelong learning. For example, using Facebook to join a group of users sharing their interests.

The concept of lifelong learning began with the worker movement. Workers were required to continue to learn throughout their working lives, in order to maintain and update their skills. Some organisations set up courses for their employees to learn from when moving from one location to another within the organisation [15].

The topic of lifelong learning using the technology was covered by Stephen Hoare in the *Guardian* newspaper [28]. The increasing use of the Web 2.0 application has shown that students like their information to be personalised and to have their own space. Their message to universities, as reported in the *Guardian* in 2007, was "Get out of MySpace".

## Chapter 2: Literature review

The Learning Experience Project is run by the Joint Information System Committee (JISC), and Lawrie Phipps, a programme manager at JISC, has stated that “students appear to want to keep their online persona private but when you ask them whether they would like instant communication with tutors or feedback on essays (via Skype or Facebook) the answer is always yes”. Phipps notes that universities use Facebook to talk to their student and he believes that Facebook will be very soon replace student unions for meeting, chatting and learning [12]. JISC’s e-learning programme manager notes that “social networking was allowing students and researches to create their own interest groups and academic communities in areas like bio-medicine”.

Gilly Salmon, professor of e-learning and learning technologies at Leicester University, has created a page on Facebook on which she posts pictures of her holiday, children and plans for the week. Her students are encouraged to visit it. Gilly states, “I am very interested in it as a potential way for students working together remotely rather than having to be in a physical space”.

The big challenge in higher education now is how to integrate social networking with already established e-learning systems.

In summary, a significant aspect of lifelong learning is that it should give a learner control over their learning. There are different applications supporting lifelong learning, for example Skype and Facebook. These are ways of keeping in touch with anyone, especially for students.

The view of PLE in this research is based on the concept of the context-aware system. The PLE is an intelligent system, which has ability to filter e-resources and meet the user’s needs based on user context i.e.; history, feedback and preferences. Moreover, it is able to predict user preferences.

### **2.3. Context-aware system**

When a computer appears to be invisible in an environment, it can be declared as being a device in a ubiquitous computing or pervasive system. The objective of context-aware system is that the system can provide intelligent, which can be used to detect context on the user behalf; this reduces the need for human attention [29]. Mark Weiser is

## Chapter 2: Literature review

considered to be the father of ubiquitous computing. He proposed ubiquitous computing in 1991 when he was a chief technologist of Xerox Palo Alto Research Centre (PARC) in the US. Weiser wrote the earliest papers on this subject area; only later were his research and projects given increased attention [30].

Ubiquitous computing in the US refers to research that originates from the ideas of Weiser, whereas pervasive systems are used more in industry and were first proposed by IBM in 1990 [29]. The main goal of pervasive systems was to use computers by distributing them in the physical environment, making them effectively invisible to the user. Hence, ubiquitous computing and pervasive systems have the same objective and the terms mean the same things. The term context-aware system was introduced for the first time by Schili and Theimer in 1994 [31]. Here they described the use of an active map service to provide information regarding a located-object; 'Located-object' refers to anything associated with physical locations such as a person, printer and workstation.

### **2.3.1. Definition of Context-aware System**

The word "context" is derived from the Latin 'con' (with or together) and 'texture' (to weave). The Macmillan English Dictionary defined context as "the surrounding which help to give meaning". Another definition can be found via the synonyms in any English dictionary, such as situation, surroundings, environment and position [32]. Many definitions of context-aware system are presented in [33, 34, 35, 36, 37, 38, 45, 55]. Some concentrate on various aspects of a context-aware system. Dey and Abowd in [38] defined a context-aware system as "a system that uses context to provide relevant information and/or service to the user, where relevancy depends on the user's task". This study has chosen Dey and Abowd's definition, because it offers a more general understanding of a context-aware system. A system that has these abilities can sense the environment and react to any changes. When applying this definition, it is shown that it has the ability to align with any context-aware system.

Loke [29] 2007 claims that context-aware systems have three basic functionalities: sensing, thinking and acting (see Figure 2.7). Sensors acquire information about the real world and, once information has been obtained using the sensors, the task to process the information and make sense of it begins. Action has to be taken following the process

## Chapter 2: Literature review

and the user of the system should retain control by cancelling, stopping or taking any action.

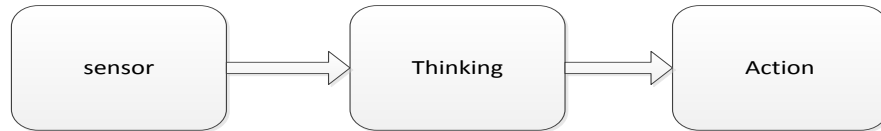


Figure 2. 7: Context-aware system functions

Kavi in [39] identifies three categories of context-aware applications.

- presenting information and services; this means presenting context information to a user or using the context to propose an action for them.
- automatically executing a service; this describes the application that triggers a command.
- attaching context information for later retrieval context; this refers to the application that tags captured data with relevant context information.

Context-aware systems can be applied to different application areas. Some of these areas are summarised below [40 and 41]:

- Airport: a notification can be sent to the passenger's mobile device, depending on the passenger's location, to inform them about shopping zones nearby and gates, arrivals and departures.
- Smart Home: using context-aware systems at home increases the quality of living and helps disabled people to live independently. It can inform appropriate institutions, such as the police and fire departments, in detecting an intruder, fire or gas leakage. In addition, it has the ability to control devices at home, for example turn lights on/off, as well as air conditioning and heaters. Furthermore, it is used to send reminders for taking medication, or notifying of emergency situations.

## Chapter 2: Literature review

- Hospitals/Healthcare: context-awareness can improve the service in hospitals, for example verifying that a nurse has carried the right medicine to patients and knowing the medical history of them.
- Leisure/Entertainment: an application is associated with the user's location. Services can be provided to a user based on where he/she is located, providing information about nearby restaurants, cinemas, theatres, etc.
- Offices: it is for the benefit of a company or institution to locate the position of their employee, as well as the status of company equipment.

### **2.3.2. Sensors**

Sensors are the base element of context-aware architecture and the whole system depends on the information that is provided by the sensors. Context-awareness takes advantage of available sensors; the main objective of the sensor is to capture information from the environment and to provide that information to the processing part in order to take an action. The fundamental element in building a context-aware system is the sensor.

A word sensor was introduced for the first time by Indulska and Sutton [42] (2003). The sensors were divided into three groups: physical, virtual and logical.

- Physical sensor:

The physical sensor is a hardware sensor that captures information about the real world, such as people and objects, etc. The physical sensor is most commonly used sensor. Most physical sensors are reviewed by [43] and summarised in table 2.2:

Type of sensor	Examples
Light	Photodiode, IR, colour sensor etc.
Visual	Cameras
Audio	Microphones
Location	Global Positioning System (GPS), Global System for Mobile Communications (GSM) and Active Badge.
Touch	Mobile devices use touch, such as with an iPhone to switch it on

## Chapter 2: Literature review

Temperature	Thermometers to detect body temperature
Air pressure	Sphygmomanometer, which is a blood pressure meter

Table 2. 2: Physical sensors

- Virtual sensor:

The virtual sensor gathers information from a virtual space. The virtual space includes software applications, operating systems and communication networks. An application such as the user's calendar can be used to locate user's activities, whereas a wireless network can detect a user's current location. An operating system can monitor users logging into a system.

- Logical sensor:

The logical sensor gathers information from both physical and virtual sensors. For example; a company detects their employees by monitoring their log-ins to PCs (virtual sensor) and by using cameras (physical sensor).

### **2.3.3. Context-aware Architecture**

In the last a few years, a number of context-aware system architectures have been proposed. Some of these architectures will be reviewed here.

The Context Broker Architecture (CoBrA) supports a context-aware system proposed in [44], as shown in Figure 2.8. CoBrA is an agent-based architecture for intelligent spaces. Intelligent spaces are meeting rooms, vehicles, etc. The architecture utilises semantic web languages to define and publish context ontology. The broker agent is at the core of the architecture, maintaining a shared model of context for all computing entities and inferring context knowledge.

## Chapter 2: Literature review

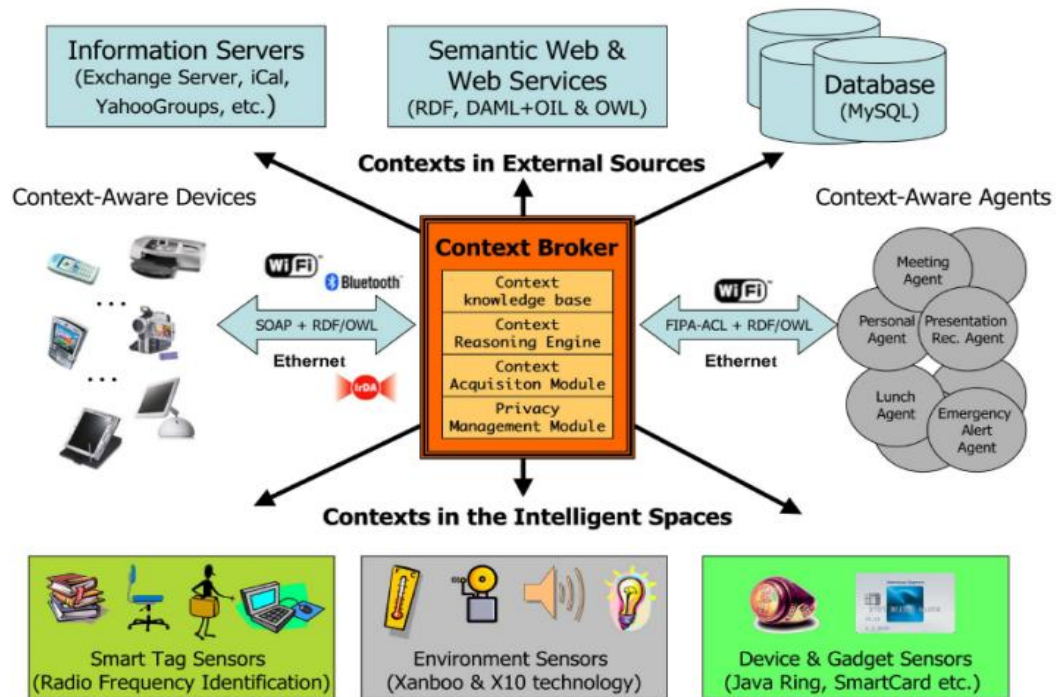


Figure 2. 8: Context broker architecture (CoBrA)[44]

[41] Proposed an infrastructure for context-aware healthcare, the architecture of which is depicted in Figure 2.9. A global healthcare system links healthcare service providers to an individual's personal and physical spaces. This system provides personalised healthcare services at the right time and the right place. The architecture consists of the following components:

- Context provider: transforms the raw context into make-ups to separate the low-from high-level context.
- Context aggregator: gathers context mark-ups from a distributed context provider.
- Context knowledge base: stores context information.
- Context reasoned: infers high-level context from basic context.
- Context query engine: allows applications to get context information from a contextual knowledge base.
- Context discoverer: support wide-area context discovery.

## Chapter 2: Literature review

- Context-aware applications: utilise the high level context information obtained from CKB.

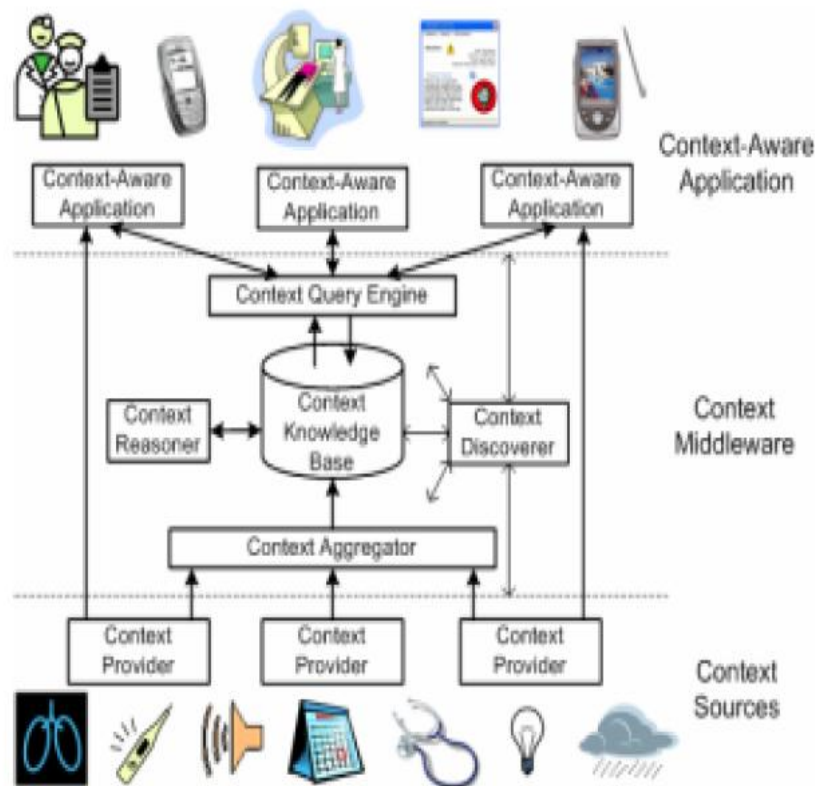


Figure 2. 9: Infrastructure for context-aware healthcare [41]

### 2.3.4. Context-aware System Development

Context-aware system development goes through the same phase as any other computer systems development. The Figure 2.10 demonstrates context-aware system development, the phases of which are: requirement definition, specification, architecture design, implementation and testing.



## Chapter 2: Literature review

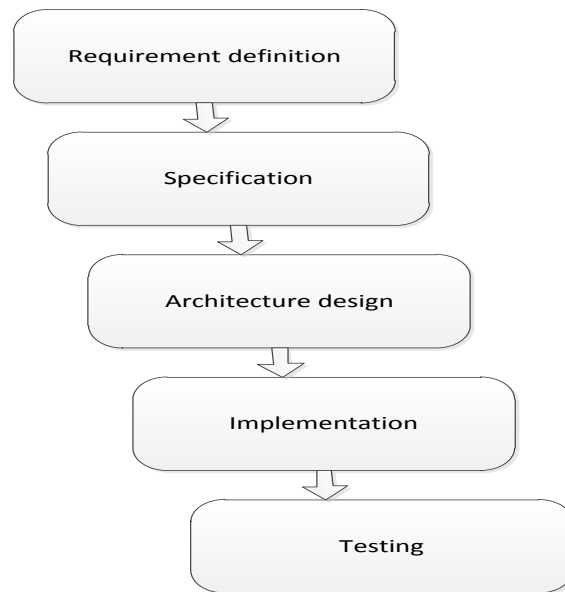


Figure 2. 10: Context-aware System Development

Requirement definition is to determine the user need, as there are some conditions and requirements for a system to be context-aware system. Yoosoo et al. [46] identified the context-aware system requirements that should support this phase.

The requirements can be specified using graphical or formal models. Graphical models are those such as the Unified Modelling Language (UML), whereas formal models use mathematical notation [47]. Some examples of formal methods include Z notation, temporal logic and the Calculus of Context-aware Ambient (CCA) [48]. Z notation is model based, whereas temporal logic is based on logic and CCA is based on process calculus. These formal methods mentioned above can have some limitations, however. For example, the Z notation does not support modelling the context-aware system. Surveys of modelling context-aware system are presented in [49, 50].

A general architecture of context-aware systems is proposed by [51]; as depicted in Figure 2.12, the architecture consists of five layers, the lowest being sensors, which include several different kinds (physical, virtual and logical).

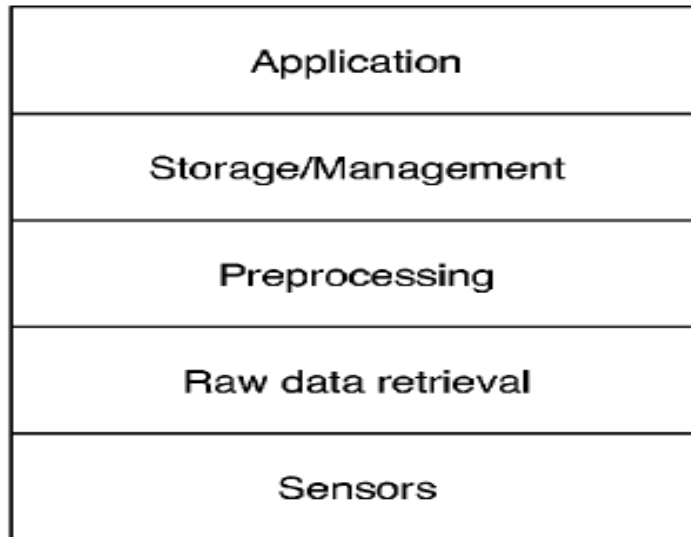


Figure 2. 11: A general architecture of context-aware system

The method of transferring the specification model to a programming language is called the implementation. The final stage is to test the new model before applying it to the real world, which looks at the behaviour of the system.

### **2.3.5. Personalised Service on a Context-aware System**

Personalisation of a context-aware system has to proactively identify the user's current needs, based on the user's current context, such as the location. Two research studies that consider the users' preferences will be reviewed, namely NAME and GUID.

NAMA (Need Aware Multi Agent) is a reminder system, which provides personalised services on context-aware systems proposed by [52]. NAMA focuses on using a user's preferences to discover their current needs and connect them to a group of services. For example, when a user is shopping and one item he/she wants to buy is near, the system will remind the user of that item.

Another proposed system based on user preferences is the GUID system [53]. The GUID system has been developed to provide city visitors with a context-aware tourist

## Chapter 2: Literature review

guide. To use the GUID system, visitors must enter their personal details and preferences into the system.

### **2.3.6. Example of a Context-aware System**

An example of a context-aware system is Active Badge. Active Badge was the first context-aware system [54] and was developed by Olivetti Research. Active Badge was developed between 1989 and 1992 at Cambridge University. The main goal of Active Badge was to provide direct location information to people in a building using sensors. Each person would wear a badge that had the ability to transmit information about the location to a central service, where a sensor network was embedded in each room of the building. Every 15 seconds, the badge would transmit information.

### **2.3.7. A summary on how the Context-aware system is important to PLE:**

A context-aware system is suitable for building PLE architecture, due to the following reasons:

- The Context-aware system is able to reduce the need for human attention; the user of the PLE does not have to change its profile regularly, because the PLE system has the ability to do that based on the behaviour of the user. If a student is in the first year of their research they are more likely to be reading books to obtain the fundamental principles relating to their topic.
- Context-aware system is adaptable according to the user's need; the PLE system will be able to predict user preferences based on their history and previous using habits.
- The Context-aware system is able to provide relevant e-resources that are important for the user; so PLE users will not receive an extortionate amount of e-resource, only that which is relevant to them and their preferences.
- Context-aware system uses sensors to take information without attention to use; the PLE system is able to gather user context from many different angles.

## Chapter 2: Literature review

### **2.4. Summary**

This chapter reviewed the literature concerning three main topics: the Virtual Learning Environment, the Personal Learning Environment and the context-aware system. Firstly, the chapter on VLEs was introduced, including the definition, the advantages and limitations of VLE. In the second part of the chapter, PLEs were described, including the definition and some examples, along with the architectures of PLE. Finally, context-aware systems were discussed, including the definition, architectures and other related information. The next chapter discusses the proposal architecture.

## **Chapter 3: Proposed architecture**

### **Objectives:**

- To present requirements and the motivation
- To propose a novel context-aware PLE architecture
- To present the stages of the proposal
- To present a case study

## Chapter 3: Proposed architecture

### **3.1. Requirement and motivation**

This study has noted the wide spread use of VLEs at all levels of education. It has also noted the limitations of VLEs and how PLEs are now being proposed to address these limitations. A review of a number of PLE architectures was presented in Chapter 2, and while they convey well the concept of a PLE, nevertheless they could best be described as high-level architectures, (sometimes referred to as frameworks in the literature), which focus mainly the functionality of PLEs. In particular, there is little published which gives a detailed designed of a PLE architecture. Moreover, the published work focus largely on the support for lifelong learning and formal / informal learning; these are two of the main limitations of VLEs.

This research argues that PLEs can support more than just learning in education, they can support users in any aspect of their lives which requires them to find / utilise e-resources available on-line. Many definitions of PLEs exist [8, 11, 12, 13]). The definition of a PLE user in this study is “any person or system who is a producer or consumer of e-resources”. This covers the use of a PLE for many purposes, in addition to traditional formal (institution-based) and informal (private, non-institution-based) academic learning. For instance, the PLE could be used for social or business purposes; examples of this type of use will be given later the chapter.

The architecture proposed in this research incorporates a context-aware engine. Thus there is intelligence built into the architecture and thus the PLE system is automatically responsive to the context information. As noted in chapter 2, there are three types of sensors in any context-aware system and these are the elements of the system that gather the context information. In this research, the emphasis will be on virtual sensors which gather the information from virtual space; virtual space here includes any systems which produce information as a set of results. Thus, the context-aware architecture and the implementation of the context-aware engine are major contributions of the work.

## Chapter 3: Proposed architecture

### **3.2 Proposal for a novel context-aware Personal Learning Environment architecture**

A proactive, context-aware PLE architecture is presented in this section and published in [105, 106]. The high-level architecture is shown in Figure 3.1. It consists of two main layers: the top layer is the PLE service, while the bottom layer is the Provider.

Top layer: PLE service

The PLE service consists of three main entities: Personal Manager, Context-Aware Engine and User Profile. These entities communicate with each other to provide the user with a service tailored to their individual needs. A user interfaces with the PLE using the Personal Manager.

Bottom layer: Provider

The Provider layer consists of various tools and independent service providers (for example Facebook, YouTube, Personal Calendar, IEEE Explore and digital libraries). These are accessed by the user through the Personal Manager component, on demand. The Provider represents the set of independent services, referred to as preferences, defined by the user and stored in the User Profile. The responsibility of the Provider is to supply the Context-Aware Engine with appropriate e-resources, in the form of virtual sensors that is consistent with the preferences in the User Profile.

There are different ways to construct the architecture. This research used two layers because it separates the PLE service from the Provider. The PLE service is able to integrate different providers.

All system components in both layers are explained in more detail below, starting with the PLE service.

## Chapter 3: Proposed architecture

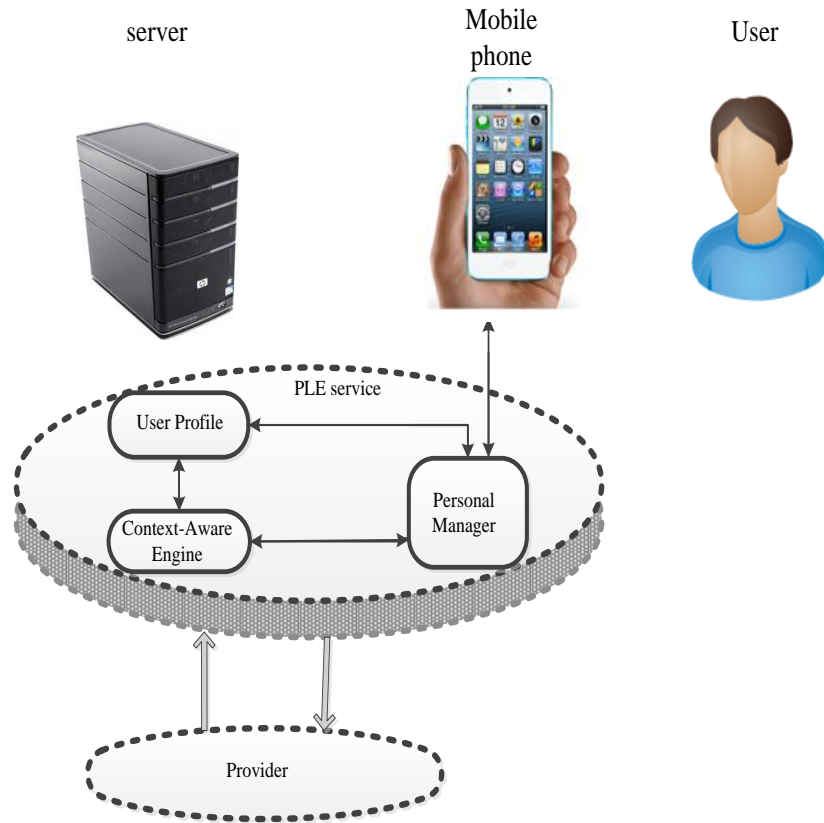


Figure 3. 1: Overview of the proposed, high-level PLE architecture

### **3.2.1. Personal Manager (PLE service)**

The only way for the user to interact with the PLE service is through the Personal Manager. It is envisaged that the Personal Manager system will be deployed as an application on standard mobile devices. The main functions of the Personal Manager are:

- To allow the user to manage his or her profile.
- To present a predication or recommendation to the user (e-resource which the proposed architecture has considered useful).



## Chapter 3: Proposed architecture

- To present to the user any new e-resources gathered by the Context-Aware Engine; this comprises a set of e-resources which are produced by providers in response to the context defined in the user profile.
- To allow the user to access independent service providers.
- To enable other users to access gathered e-resources from the user profile via a public portal.
- To allow the user to give their feedback (select /reject the e-resources found by the system).

In summary, the Personal Manager will coordinate all of the user's e-communications.

### **3.2.2 Context-Aware Engine (PLE service)**

The Context-Aware Engine subsystem is responsible for filtering and interpreting the e-resources (for example, virtual sensor information) produced by the tools or services of the bottom Provider layer. This filtering is performed using the Context History together with the preferences in the User Profile information and user feedback provided by the user. This ensures that the user receives only relevant e-resources. Figure 3.2 shows the components of the Context-Aware Engine: Acquisition, Context History and Context Reasoner.

## Chapter 3: Proposed architecture

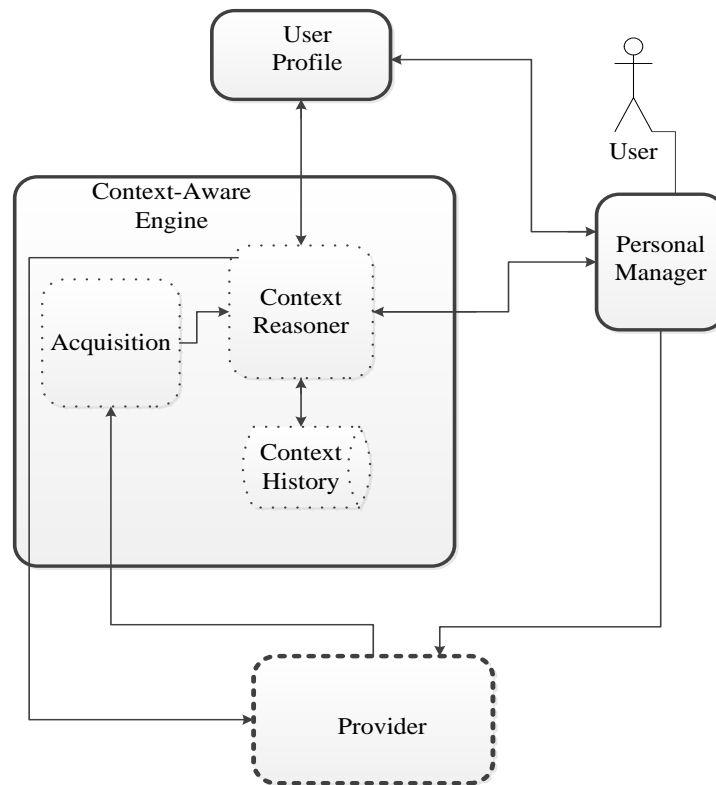


Figure 3. 2: Context-aware Engine subsystems

### **Acquisition component**

The main purpose of the Acquisition component is to gather or isolate e-resources (via virtual sensors), originating from the independent service providers in the bottom layer of the architecture, and send it to the Context Reasoner. The Acquisition component must therefore have an API to interact with the independent service provider.

### **Context History component**

The Context History component is responsible for the storage of high-level e-resources that has been gathered previously and has thus already been delivered to the user. The storage of this information is essential as it prevents duplicate information being sent to the user at a later point in time. Furthermore, Context History will help improve the

### Chapter 3: Proposed architecture

Context Reasoner functionality by providing a more detailed history of the user's previous selections.

#### **Context Reasoner subsystem**

The Context Reasoner subsystem is responsible for using context information (user information) and e-resources (produced by a provider) in an intelligent way and is the most challenging function of the Context-Aware Engine. Fig 3.3 shows the functionality of the Context Reasoner and its role within the Context-Aware Engine in providing the PLE service.

The overall responsibility of the Context Reasoner is to extract relevant high-level e-resources from the raw data (e-resources) using the information stored in the Context History, the User Profile and user feedback.

In this approach, the Context Reasoner subsystem consists of the following components: Detector, Profile Detector, Generator and Generic subsystem. The components of the Context Reasoner are described as follows:

### Chapter 3: Proposed architecture

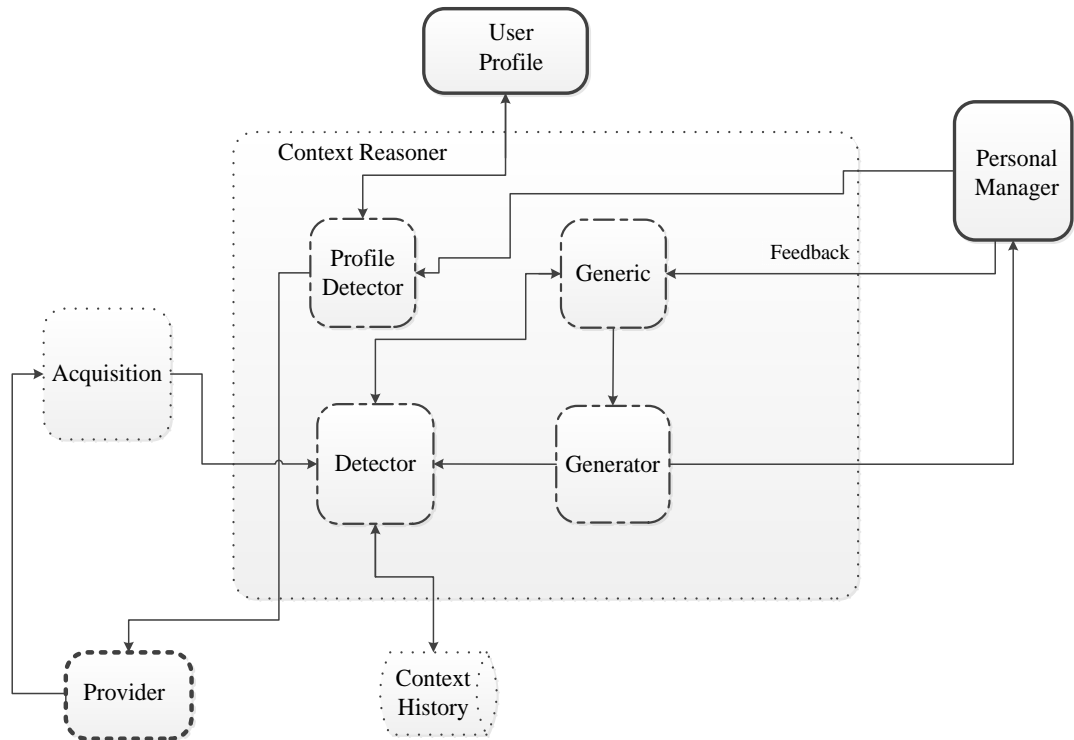


Figure 3. 3: Context Reasoner

- **Detector**

The Detector continually receives e-resources from Acquisition. It uses the Context History to determine whether this is new e-resources that has not been previously delivered to the user. If the e-resource is not new, the Detector is directed to stop processing; otherwise, it continues. Thus, the Detector must contain an underlying mechanism that allows it to recognise the context.

- **Profile Detector**

The Profile Detector is connected to the user profile. It is responsible for providing both the Provider and the Generic components with information about the user.

- **Generator**

The Generator is the component that creates or produces an action based on input from the Generic. The action should be delivered to the user via the Personal Manager. For instance, the action may send information to the Personal Manager to notify them that a new photograph of interest to the user has been posted on Flickr. Other functionalities for the Generator include updating the context history with details of the action, via the Detector.

## Chapter 3: Proposed architecture

- **Generic subsystem**

The Generic subsystem function predicts or suggests new user preferences based on existing context information stored in the Context History, together with information stored in the User Profile and user feedback. This function therefore has the ability to learn from the habits of the user. The other responsibility of the Generic function is to check that a similar e-resource has not previously been rejected by the user.

Amazon recommendations are a good example of using context history. When a customer purchases a book from Amazon, the website recommends a similar book the next time he or she logs in.

The Generic subsystem consists of five main elements, as shown in figure 3.4: Analysis Feedback, Accepted, Rejected, Prediction and Similarity. These components are described below:

### Chapter 3: Proposed architecture

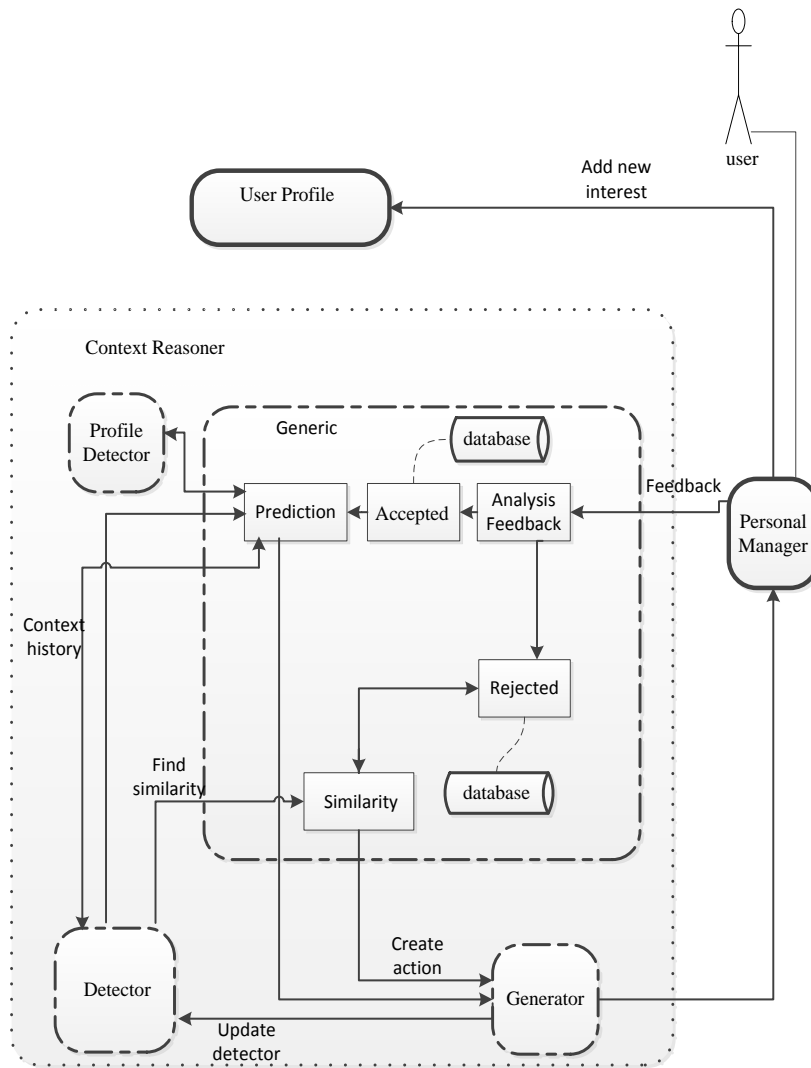


Figure 3. 4: Generic

#### **a) Analysis Feedback component**

The Analysis Feedback component is introduced to increase the precision of the user requirements. This component is responsible for distributing the feedback depending on the rank level that has been received from the user of the system. There are five ranking levels ranging from irrelevant through to very relevant. These ranking are set to update the Accepted and Rejected components.

#### **b) Accepted**

This component stores all the e-resources ranked highest by user feedback. The results from the Accepted component will be fed into the Prediction component in order to

## Chapter 3: Proposed architecture

make a recommendation to the user. This component has 4 rank levels - Fine, Of interest, relevant and Very Relevant.

### **c) Rejected**

This component contains all the declined ranks, i.e. those the user has ranked 'irrelevant'. This means that the user's feedback constitutes rejection. Consequently, in the future, this component will help to prevent the receipt of similar e-resource which has previously been rejected. It helps the system to discover unsolicited information.

### **d) Prediction**

The Prediction component has the functionality to predict/produce contextual information. It is able to learn, reason and be dynamic. It is an automated refinement of the user preferences. The Prediction component has the ability to make predictions based on the contextual information, which it receives from three components: Profile Detector, Detector and Accepted. For example, the system is able to recommend or suggest some providers to the user. The results from this component will be delivered to the Generator. In order to implement the Prediction component, a Bayesian network, which is an artificial intelligence technique, will be used; this is discussed in chapter 4 and chapter 5.

### **e) Similarity**

This component will prevent any form of repetitiveness occurring and without informing the user, will automatically reject and remove anything an e-resource that has been rejected in the past. For example, if a user has rejected an e-resource called "Intelligent System", in the future the system will prevent this e-resource.

### **3.2.3. User Profile**

The User Profile contains information about the user and is comprised of four elements: Personal or Static Information, Preferences, User Database and Profile Manager, as shown in Figure 3.5. The user must be able to define policy rules regarding the use of his or her profile by a third party. This gives the user the ability to decide when, with

### Chapter 3: Proposed architecture

whom and what information they are willing to share. The components of the User Profile are explained as follows:

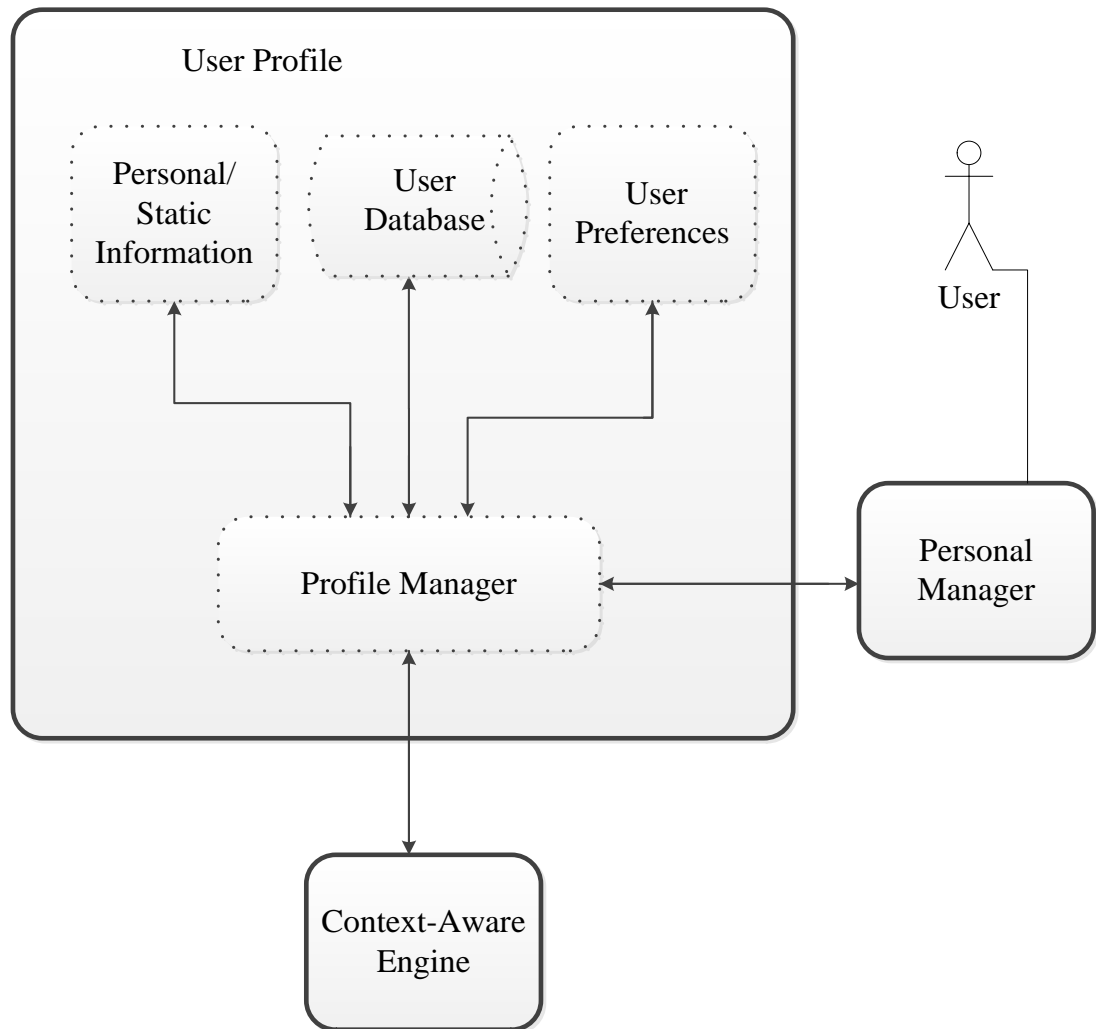


Figure 3. 5: User Profile

#### **a) Profile Manager**

The main aim of the Profile Manager is to provide the Context-Aware Engine with the user's personal or static Information and preferences. The user has the ability to access his or her profile to create, add and delete information through the Personal Manager.



## Chapter 3: Proposed architecture

### **b) Personal or Static Information**

Personal or Static Information includes standard information, such as name, address, gender, marital status and email address.

### **c) User preferences**

The user can provide information relating to his or her various interests, noted here as User Preferences. It is expected that the user's preferences will change over time and that the user will be able to modify them directly. As an example, Jenny is a PLE user whose preferences state that she is only interested in being told about new family photographs posted on Facebook and Twitter. The PLE system will therefore not alert Jenny to any other activity from these independent service providers via the Context-Aware Engine

User preferences have different lifetimes associated with them. For example, a user may be interested in a preference for a short time, which will then naturally time out and be removed; students can be given an essay and require resources relating to the essay topic: the requirement for resources on the essay topic is an example of a short-term preference. In contrast, the user may have a lifelong interest in a certain subject and, while the interest might evolve, it will remain in the profile until explicitly deleted by the user.

In addition, User Preferences accept suggestions about changing preferences from the Generic function within the Context-Aware Engine. User preferences are affected by the user's reaction to gathered information; for example when a user deletes information, the Generic subsystem can alert or change the user preferences to create a dynamic phase that will provide better results in the future.

### **d) User Database**

The User Database comprises e-resources to which the user wishes to have long-term access. These are stored in the user database and are organised by the user.

## **3.3. A walk-through of the architecture**

For most systems, there are typically a number of major functions that either the user can choose to execute explicitly (for example, a function chosen from a menu) or which

## Chapter 3: Proposed architecture

execute ‘on the fly’. In this walk-through these functions are referred to as stages and a walk-through of each stage is presented. Three major functions (stages) are identified: explicit system registration by the user, detection of context and the generic ‘on the fly’ by the system.

### **3.3.1. Stage 1: system registration**

When a user logs on to the PLE service via their Personal Manager for the first time, the user has to register and thereby create their user profile.

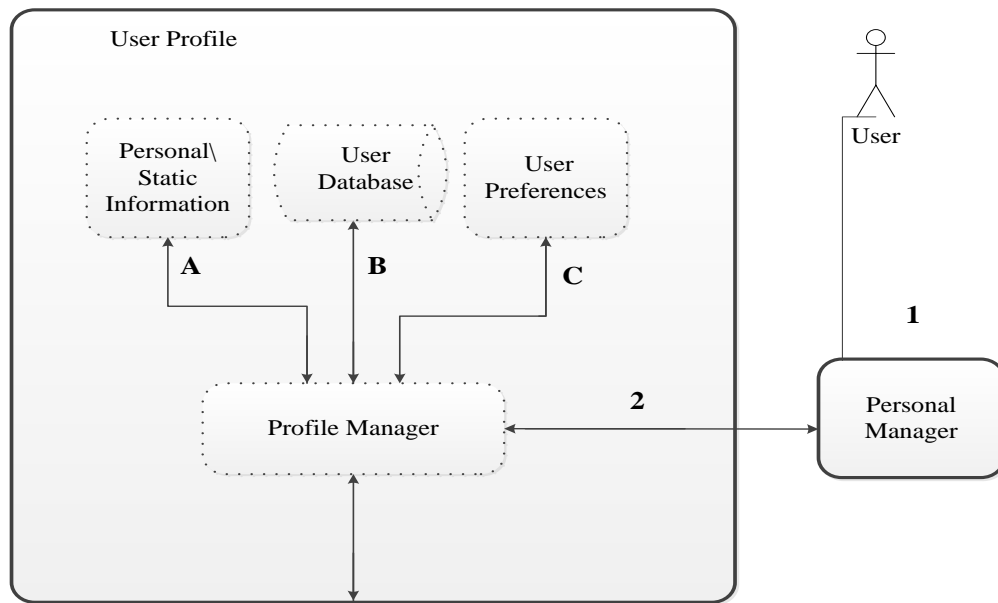


Figure 3. 6: Registration stage

There are two steps to complete the registration stage as are shown in Figure 3.6, and these are as follows:

**Step 1:** The only way for the user to interact with this system is through the Personal Manager. The Personal Manager will typically be deployed as an application on mobile devices.

**Step 2:** The Personal Manager (the software in the user’s mobile device) interacts with the Profile Manager and allows the user to manage (create, edit & delete) their own profile. The profile information is distributed across 3 logical databases: Personal Information, User Database and Preferences. An explanation of these is given below.

## Chapter 3: Proposed architecture

### A. Management of User's Personal Information

It is mandatory for the user to supply his/her personal information when they use the PLE for the first time. For example, if we suppose that the user is a research student, the personal information will typically include their first name, last name, gender, university name, course details, student ID number, date of birth and email address. This type of information is static in that it changes infrequently.

### B. Management of User Preferences

A user may choose to enter their preferences into the system. The user may have a number of interests and it is expected that these will change and evolve over time. For example, a PhD student typically begins with a broad, general interest in a topic, and gradually refines their interest as more knowledge is gained. Thus their preferences need to change to reflect their growing knowledge. The system allows users to add, modify and delete their preferences at any time. For instance they may define their favourite providers such as IEEE, ACM and Science Direct as preferences.

In addition, the user must be able to accept recommendations about changing their preferences; these recommendations will originate from the Generic function within the Context-Aware Engine. For example, the system can detect that a student is interested in seeing a survey paper on e-learning. It is expected that recommendations on possible resources of interest to the user will be a continual process.

### C. Management of User Database

The User Database comprises resources that the user wishes to have long-term access to. These are stored in the user database and can be organised by the user. An example of this could be that the research student can store material for a topic which they are no longer actively researching.

### **3.3.2. Stage 2: Detection of context**

There are six steps in the detection of context stage. Most of these steps do not deal directly with the user but are automatically invoked by other system PLE components. These steps are explained as follows and the order of execution shown in Figure 3.7 :

### Chapter 3: Proposed architecture

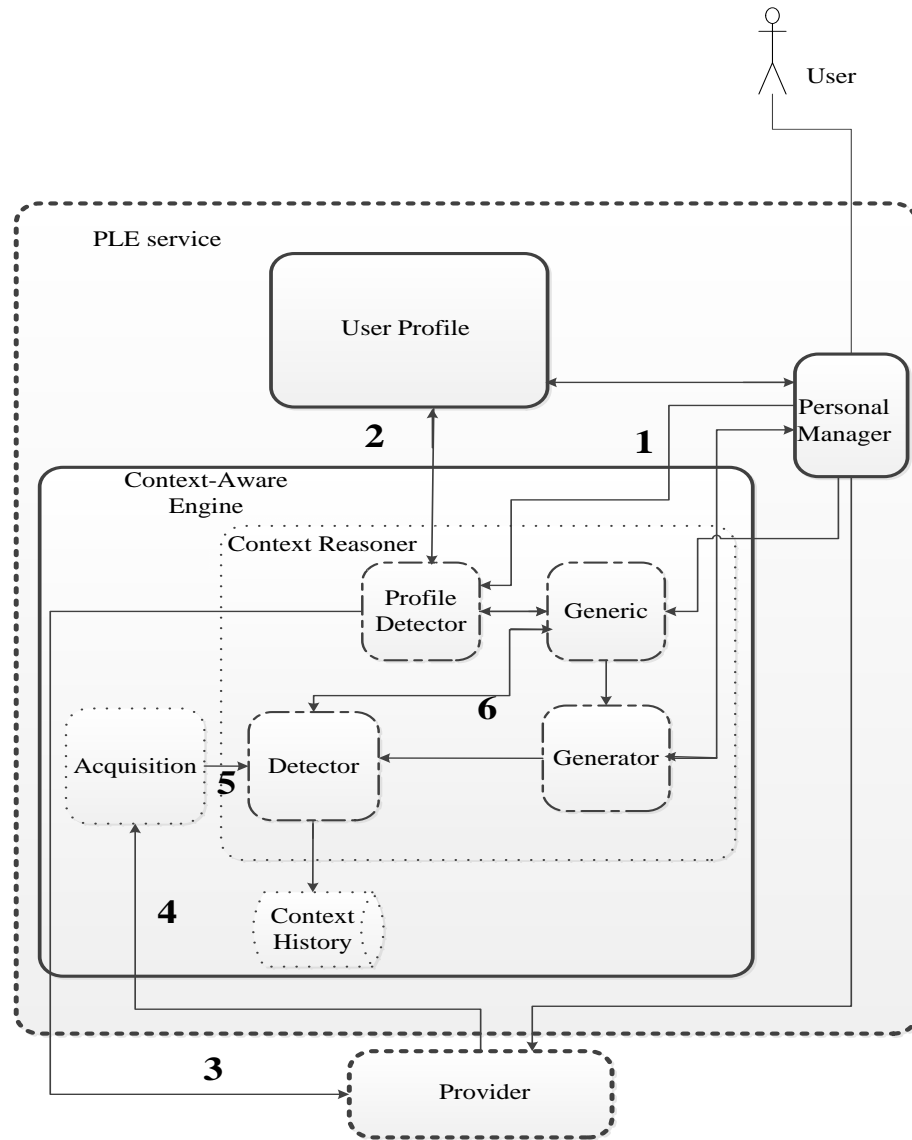


Figure 3. 7: Detection of context stage

**Step 1:** Providing a user profile is in place, the user can choose a ‘search’ option; this is available on the Personal Manager. This search request will automatically invoke the Profile Detector.

**Step 2:** The Profile Detector will then interact with the User Profile in order to gather user preferences (context).

**Step 3:** having gathered the relevant user preference information the Profile Detector will send this a request for the e-resources identified by the preferences to Provider.

### Chapter 3: Proposed architecture

**Step 4:** Using the information passed from the Profile Detector, the Provider, typically an independent service, will initiate a search for the requested e-resources; the results of this search will be forwarded to the Acquisition component. For example, the Provider for a research student could collect e-resources from different providers such as IEEE Explore, ACM and YouTube.

**Step 5:** Once Acquisition has received the e-resources from the Provider, then it can forward the gathered e-resources to the Detector.

**Step 6:** Once the Detector receives e-resources from Acquisition, its function is to determine whether this is a new e-resource (knowledge) that has not previously been delivered to the user. Assuming that the e-resources is new for the user, it should be sent to Generic. By the end of this step, stage 2 is completed, and next stage will be the Generic stage.

#### **3.3.3. Stage 3: Generic stage**

The last stage is the Generic stage. The Generic stage is invoked when it receives e-resources from the Detector; the steps in this stage are shown in Figure 3.8:

### Chapter 3: Proposed architecture

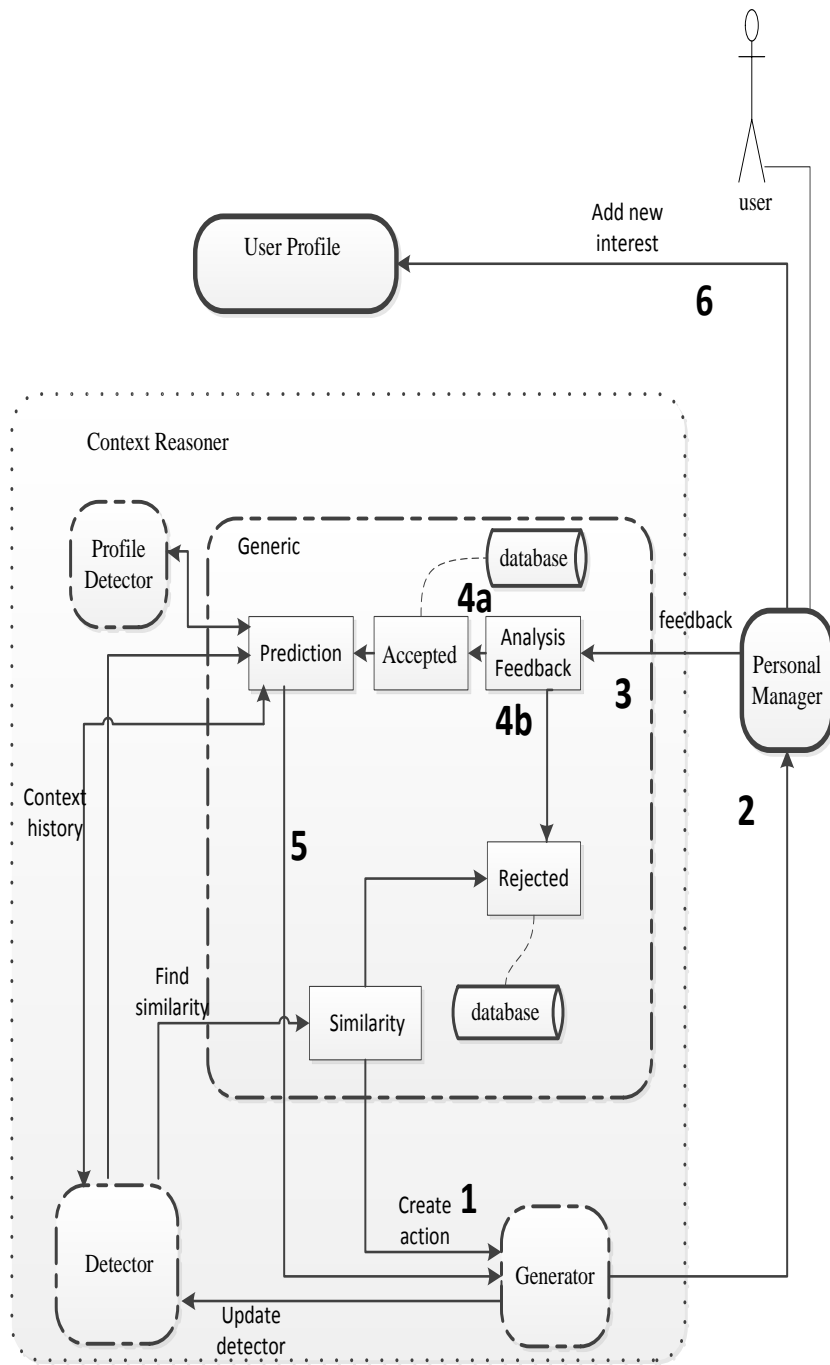


Figure 3. 8: Generic stage

**Step 1:** The Similarity component is responsible for checking that similar e-resources have not been rejected before. For example, suppose that a user has

### Chapter 3: Proposed architecture

rejected e-resources with the title “Modelling uncertainty in a context-aware system”. If the Detector component forwards a further resource entitled “Using fuzzy logic in a context-aware system” to the Similarity component this e-resource will be rejected automatically. Assuming the new e-resource does not match any user rejections, a message should be sent to the Generator in order to create the action.

**Step 2:** The Generator is the component that creates or produces a message based on its input. The message should be delivered to the user via the Personal Manager.

**Step 3:** The Personal Manager receives external feedback from the user based on the e-resources which has been delivered. This feedback will indicate how useful the e-resource is to the user. For example, after a student has evaluated the e-resources (read a conference paper) they will then be in a position to rank the usefulness of the e-resources from irrelevant through to very relevant.

**Step 4:** Analysis Feedback is introduced to increase the precision of the user preferences. This component has a policy that analyses the user feedback, and the outcome must be one of the following:

- a) The outcome is considered relevant only if it is useful to the user.
- b) If the e-resources is considered irrelevant to the user, the feedback reflects this; therefore the system will not deliver similar e-resources to the user. This feedback is often introduced to increase the accuracy of a system by correcting unwanted information. Thus the results will go to the Rejected component.

**Step 5:** The Prediction component uses artificial intelligence (AI) to predict what specific preferences user might prefer. Bayesian networks have been chosen as the most suitable AI technique to use, based on the intelligence requirements of the PLE; the case for the use of Bayesian networks is argued in chapter 5. The Prediction component requires information from three components: Accepted, Detector and Profile Detector. When a preference is predicated, the result of this

## Chapter 3: Proposed architecture

step should be sent to the Generator, which is responsible for creating the action and delivering it to the user via the Personal Manager.

Ultimately the user has to take the final decision to accept or reject the new interest, via the Personal Manager.

**Step 6:** Assuming that the user has accepted the predication of the system, the User Profile must be updated accordingly. This step will be automatic once the user has accepted the predication from the system.

### **3.4. Advantages of the proposed context-aware PLE architecture**

The advantages of the context-aware PLE architecture presented in this chapter are:

- The system is responsive to each individual user, on demand.
- There is an integration of both informal and formal learning.
- The system is able to predict user preferences or interests.
- The system can prevent duplicate e-resources (e.g. an e-resource being presented to the user more than once) being sent to the user at a later point in time.
- The user is able to update the system by giving feedback to their PLE system in order to improve the result over time.
- The proposed systems will not only support e-resources for learning but other life aspects, i.e. business and social aspects.
- The system supports personalisation; the user of the system is able to select the required tools and ignore irrelevant resources.
- This is a two-way learning system, giving the user the ability to share and upload related information to other users via their PLE interface.
- The system is open to cover any provider.

### **3.5. Limitations of the context-aware PLE architecture**

This section presents the most important limitations inherent within the context-aware PLE architecture. As for any proposal, the proposed architecture in this research may have certain drawbacks when applied to a real system.



## Chapter 3: Proposed architecture

- There is one major limitation in the system, which is that the Detector requires a great deal of information to be filtered (overload risk). The Detector's purpose is to continually receive e-resources from acquisition, and its function is to check whether this new e-resource has not been previously delivered to the user. Accordingly, the processing time of the Detector component will be slow.
- Another limitation of the proposal is that the system will take time to acquire e-resources from the various providers.
- Storage for the Context History is another issue. As the Context History component is responsible for the storage of the high-level e-resources that have been gathered previously, and has thus already been delivered to the user, a large quantity of information must be stored in the context history. This is one of the drawbacks of the system, as the memory for this component must therefore be large.

### **3.6. Implementation Considerations**

It would take many man years to fully implement the PLE architecture proposed in this chapter, as the PLE application is essentially a web 2.0 type application, with a GUI similar to something like Facebook. This is clearly not possible within the timescale of a PhD. What sets the PLE application apart from the majority (all) of typical web 2.0 applications is that it is context aware. The context aware subsystem requires the support of an embedded artificial intelligence component – the Prediction component shown in the Generic subsystem. Thus it is proposed to focus on implementing this Prediction component as this is the major challenge that is unique to the PLE application, compared to other web 2.0 applications and other PLE architecture.

### **3.7. Case study**

#### **3.7.1. Introduction**

In this section, a case study for the proposed architecture is presented. The definition of PLE user in this research is: "Anyone who is a producer or consumer of e-resources". This covers the use of PLE for many purposes such as: business purposes, social

## Chapter 3: Proposed architecture

purposes or academic purposes. In this instance, the case study in this research will refer to the academic aspect. The case study will show how the proposed architecture works in real life.

### **3.7.2. Jenny, PhD student**

Jenny is a postgraduate student who finished her Masters last year and has recently started working towards her PhD (January 2013); she studies in the Faculty of Technology.

### **3.7.3. Requirements study of Jenny:**

Jenny will use the PLE application proposed in this research to gather e-resources relating to her PhD. In particular, she requires the application to:-

- Allow her to define her preferences, and maintain these, on demand.
- Suggest good sources for the type of e-resources she needs
- Suggest an e-resource once only for a given search criteria, regardless of how many times she carries out that search. Jenny will not receive same e-resource more than one.
- Allow her to provide feedback in the form of a ranking so that the PLE application can learn which e-resources Jenny finds the most useful and use this knowledge in future searches.

### **3.7.4. The interface of PLE application**

Jenny interacts with the PLE application through the Personal Manager which is shown in Figure 3.9. The Personal Manager is an application which Jenny can deploy on her mobile devices, such as the iPhone and laptop.

### Chapter 3: Proposed architecture

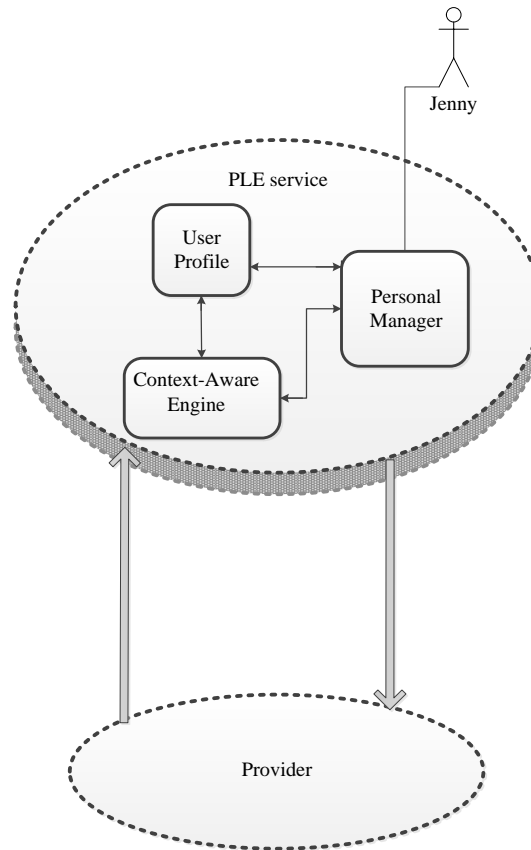


Figure 3. 9: Jenny interacts with a context-aware PLE

The Personal Manager application can offer the following features for Jenny:

- Allow Jenny to manage her profile.
- Present a predication or recommendation (e-resource which the PLE application has considered useful) to Jenny.
- Present to Jenny any new e-resources gathered by the Context Reasoner defined in the profile.
- Allow Jenny to access independent service providers.
- Enable other users (authorised by Jenny) to access gathered e-resources from Jenny's profile via a public portal.
- Allow Jenny to give their feedback (select /reject the resources found by the system).

### Chapter 3: Proposed architecture

Figure 3.10 shows an example main page that the Personal Manager may present to Jenny. It is the top level menu in the Persona Manager application and allows access to Jenny's Preferences, Predicate Preferences, Third party and Feedback. The functionality of each of these menu items are explained below.

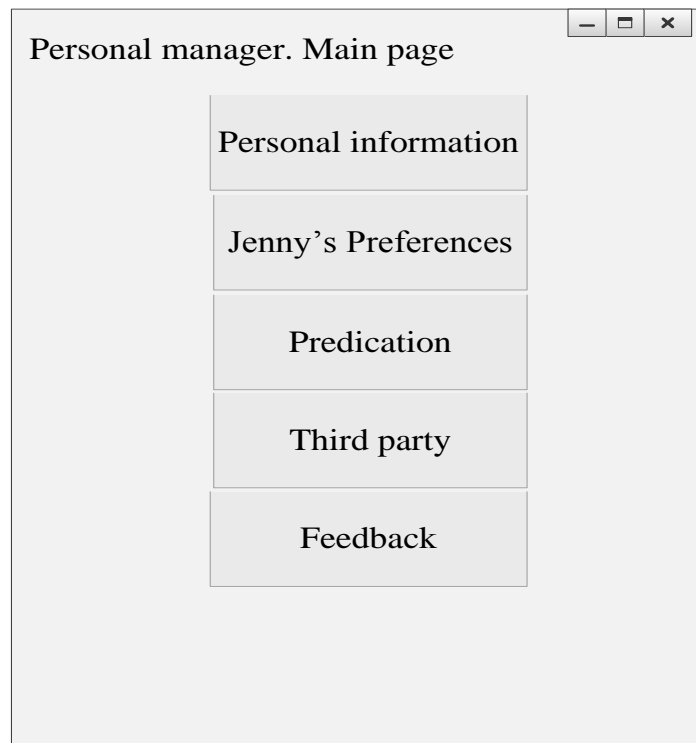
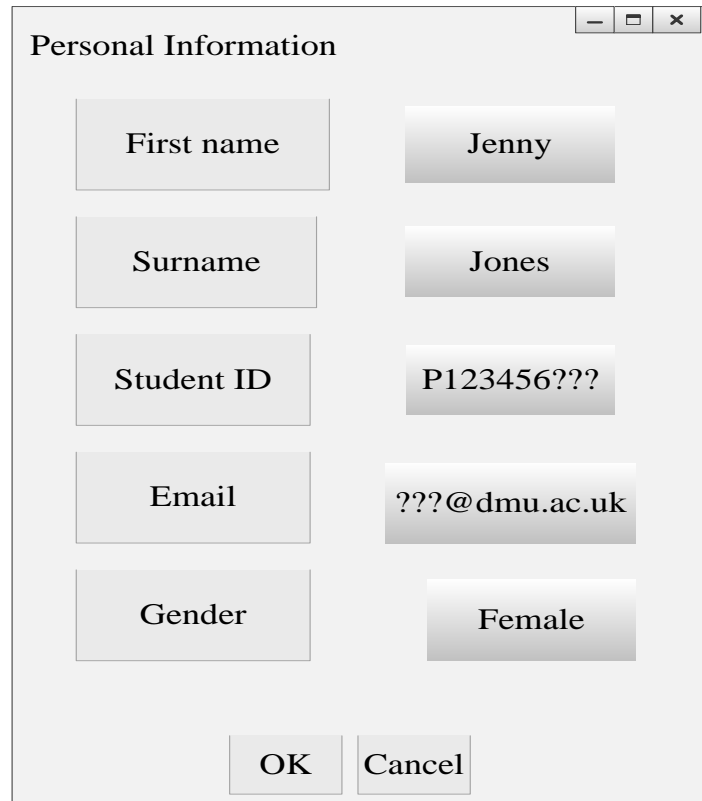


Figure 3. 10: Jenny's Personal Manager

More details about the main page discuss as follows.

- **Personal Information:** Jenny has to register with the PLE application before using it. Figure 3.11 shows typical information Jenny will be expected to provide for the Personal Information Link.

### Chapter 3: Proposed architecture



Personal Information	
First name	Jenny
Surname	Jones
Student ID	P123456??
Email	???@dmu.ac.uk
Gender	Female
<div>OK Cancel</div>	

Figure 3. 11: Personal Information details

- **Jenny Preferences:** Jenny's preferences include defining Providers (of e-resources) and Keywords relating to her PhD subject area. The Provider refers to the providers of published work, i.e. databases such as the IEEE. There are two ways for Jenny to select her providers, Figure 3.12 presents these. The first one is to select the provider directly herself, the second way is to use the system to make predictions or recommendations as to which provider she might use. Figure 4.13 shows three provider were selected by Jenny: IEEE, British Library and ACM.

## Chapter 3: Proposed architecture

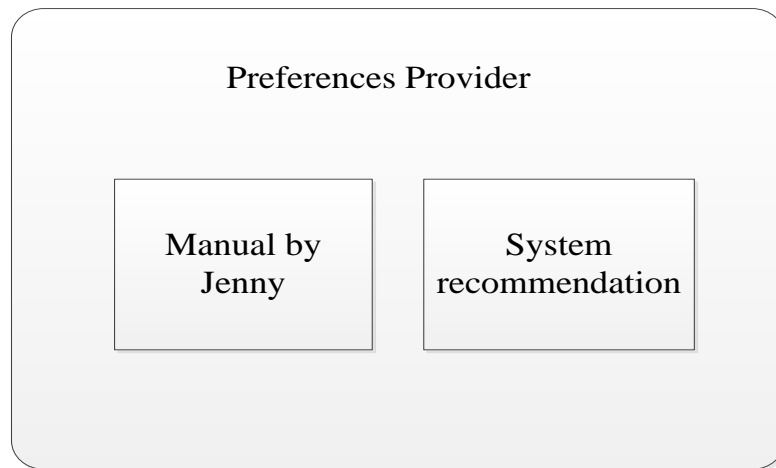


Figure 3. 12: Ways to select providers

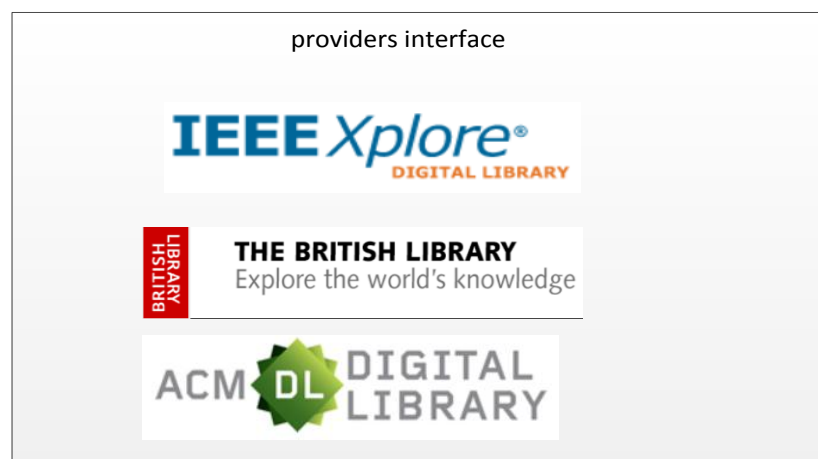


Figure 3. 13: Three different providers

- **Predicted preferences:** The system is able to propose providers such as IEEE, British Library and ACM, based on information, which Jenny should input to the system. The system uses artificial intelligence techniques that are able to predict suitable providers for Jenny to use. Figure 3.14 shows Jenny an interface that allows her to insert her information about provider preferences. Whereas Figure 3.15 shows Jenny an interface that allows her to insert her keyword preferences from the

### Chapter 3: Proposed architecture

computer science vocabulary. Chapter 5 will show that how the predicted preferences will be implemented.

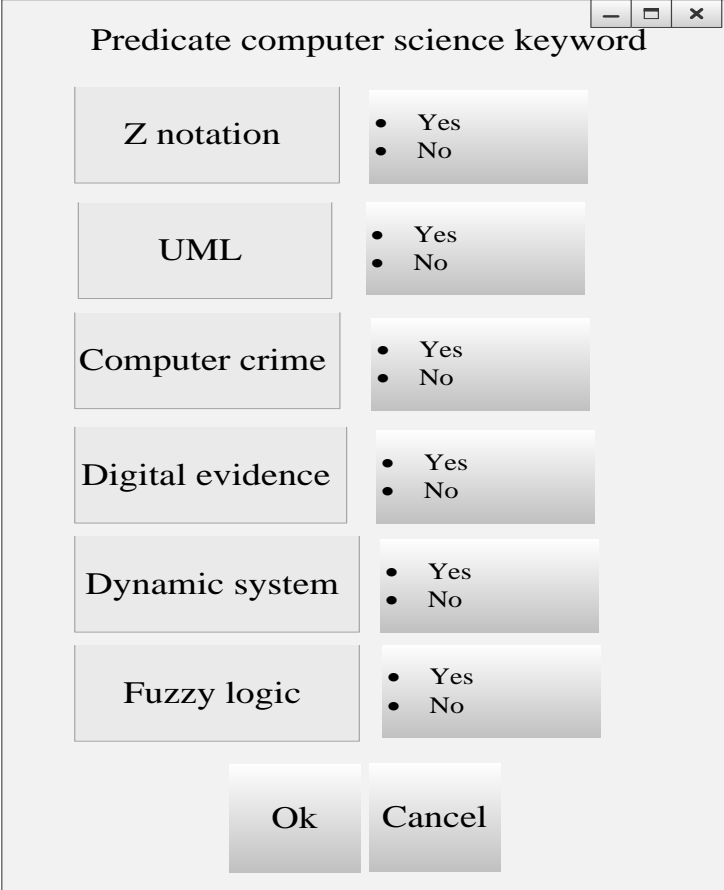
Predict provider

Cost	<ul style="list-style-type: none"><li>• Yes</li><li>• No</li></ul>
Detalis	<ul style="list-style-type: none"><li>• Yes</li><li>• No</li></ul>
General	<ul style="list-style-type: none"><li>• Yes</li><li>• No</li></ul>
Supervisor advice	<ul style="list-style-type: none"><li>• Ethos</li><li>• Library</li><li>• IEEE</li></ul>
Year of study	<ul style="list-style-type: none"><li>• First</li><li>• Second</li><li>• Third</li></ul>

Ok Cancel

Figure 3. 14: Predict provider

### Chapter 3: Proposed architecture



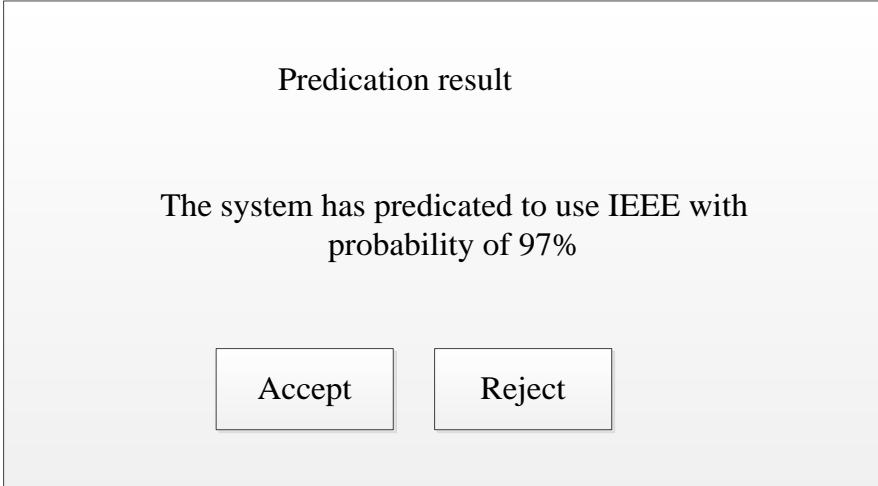
Predicate computer science keyword

Z notation	<ul style="list-style-type: none"><li>• Yes</li><li>• No</li></ul>
UML	<ul style="list-style-type: none"><li>• Yes</li><li>• No</li></ul>
Computer crime	<ul style="list-style-type: none"><li>• Yes</li><li>• No</li></ul>
Digital evidence	<ul style="list-style-type: none"><li>• Yes</li><li>• No</li></ul>
Dynamic system	<ul style="list-style-type: none"><li>• Yes</li><li>• No</li></ul>
Fuzzy logic	<ul style="list-style-type: none"><li>• Yes</li><li>• No</li></ul>

Ok Cancel

Figure 3. 15: Predicate computer science keyword

The Prediction component of the architecture will process the inserted information and deliver a result to Jenny, as shown in Figure 3.16.



Predication result

The system has predicated to use IEEE with probability of 97%

Accept Reject

Figure 3. 16: Predication result



### Chapter 3: Proposed architecture

- **Third party:** The PLE application allows Jenny to interact with other users, see Figure 3.17. Thus Jenny must also be able to define access policy rules regarding the use her profile by a third party. This gives Jenny the ability to decide when, with whom and what information she is willing to share. The system is a two way learning system. This gives Jenny the ability to share and upload related information to other users via their personal manager page. For example, James is also a PhD student; he would like to receive details of the journal papers that Jenny has referenced from ACM.

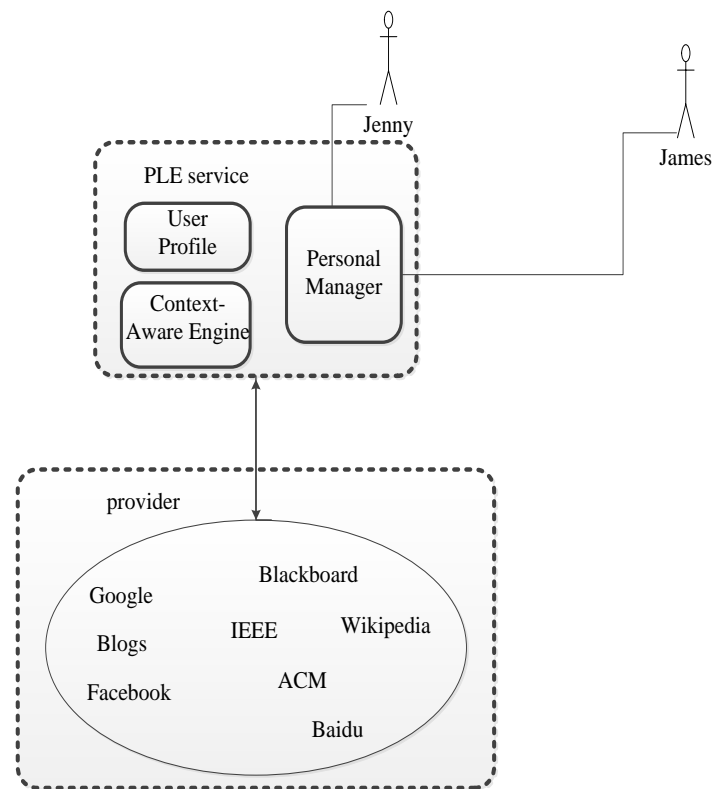


Figure 3. 17: Third party

- **Feedback:** The PLE application requires the user to give feedback on the e-resources that have been found in the sources defines in the user profile. The interface for feedback consists of the e-resources and at the bottom provides

## Chapter 3: Proposed architecture

space to rank that paper. The ranking starts with very relevant and continues through to irrelevant. The feedback is very important for the system, in order to learn, and thus refine, what e-resources Jenny finds useful. Figure 3.18 shows the feedback interface.

2009 International Conference on Advanced Information Networking and Applications Workshops

CCA: a Calculus of Context-aware Ambients

François Siewe, Antonio Cau and Hussein Zedan  
Software Technology Research Laboratory  
De Montfort University  
Leicester, United Kingdom  
Email: {fsiewe, acau, hzedan}@dmu.ac.uk

Abstract

We present a process calculus, CCA, for the modelling and verification of mobile systems that are context-aware. This process calculus is built upon the calculus of mobile ambients and introduces new constructs to enable ambients and processes to be aware of the environment in which they are being executed. This results in a powerful calculus where mobility and context-awareness are first-class citizens. We present the syntax and a formal semantics of the calculus. We show that CCA can encode the  $\pi$ -calculus, and illustrate the pragmatics of the calculus through a case study of a context-aware hospital bed.

**Keywords.** Context-awareness, process calculus, mobile ambient, pervasive computing

1. Introduction

Pervasive computing is a new paradigm for next-generation distributed systems where computers disappear in the background of the users everyday activities. In such a paradigm computation is performed on a multitude of often invisible small devices interconnected through a wireless network. Fundamental to pervasive computing is that any component (including user, hardware and software) can be mobile and that computations are context-aware. As a result, mobility and context-awareness are important features of any design framework for pervasive computing applications.

are first-class citizens. Our contributions are summarised as follows:

- we propose a logical language for expressing properties of the contexts of CCA processes (Sect. 3). We call a formula in this logic a *context expression*. Context expressions are used in CCA to guard capabilities so that they are performed only in specified contexts. A *context-guarded capability* has the form  $\kappa!M$  where the guard  $\kappa$  is a context expression and  $M$  is a capability. Such a capability is performed only when the environment satisfies its guard. We give the semantics of context expressions in terms of a satisfaction relation with respect to a formal model of contexts based on the hierarchical structure of ambients.
- we give the syntax (Sect. 2) and formal semantics (Sect. 4) of CCA which extends CBA with the following features: (i) *context-guarded capability* as explained above; (ii) *process abstraction* as a mechanism for context provision; and (iii) *process call* as a mechanism for context acquisition.
- we show that CCA can encode the  $\pi$ -calculus (Sect. 5), and illustrate the pragmatics of the calculus through a case study of a context-aware hospital bed (Sect. 6).

2. Syntax of Processes and Capabilities

This section introduces the syntax of the language of CCA. As in the  $\pi$ -calculus [3], [4], the simplest entities of the calculus are *names*. These are used to name ambients, locations

Feedback

Irrelevant

Fine

Of interest

Relevant

Very relevant

Submit

Figure 3. 18: Feedback

## Chapter 3: Proposed architecture

### **3.7.5. Scenarios**

This section presents two different scenarios for the case study.

#### **First scenario:**

This section demonstrates the first scenario; there are certain steps to follow for this scenario:

Step 1:

Jenny has the ability to provide the PLE application with any information relating to her various interests. For example, Jenny enters “Modelling and reasoning about uncertainty in context-aware systems” into her PLE application. She wishes to gather e-resources relating to this interest from different providers, e.g. IEEE, ACM and Science Direct.

Step 2:

The PLE is able to search and gather e-resources from different providers. The result of the search is four e-resources, as shown in Table 3.1.

No.	E-resource name
1	Fuzzy logic
2	Bayesian Networks
3	Neural networks
4	Probabilistic logic

Table 3.1: Result step 2

Step 3:

The PLE application must determine whether the e-resources are new and have not previously been delivered to Jenny. Indeed, Jenny has read a paper entitled “Probabilistic logic”, and accordingly, the system ceases processing this e-resource. However, the three other e-resources continue to be processed. Thus, the result of this step is three e-resources, as shown in Table 3.2.

### Chapter 3: Proposed architecture

No.	E-resource name
1	Fuzzy logic
2	Bayesian Networks
3	Neural networks

Table 3.2: Result step 3

Step 4:

The PLE application is also responsible for preventing any similar e-resources that have previously been rejected by Jenny. It is assumed that Jenny has previously rejected an e-resource called “Introduction to Neural networks”, and therefore, the paper entitled “Neural Networks” is prevented by the system and its process is discontinued.

The PLE application is able to post the two other e-resources to Jenny via her PLE application interface. The PLE application is able to provide the names of the e-resources, as shown in Table 3.3.

No.	E-resource name
1	Fuzzy logic
2	Bayesian Networks

Table 3.3: Result step 4

Step 5:

Once Jenny has received the e-resources from the PLE application, she can read them. Then, the PLE application requires Jenny to give feedback on the e-resources that she has received. The feedback is given as a ranking level, and the 5 options available to Jenny are ‘very relevant’, ‘relevant’, ‘of interest’, ‘fine’ and ‘irrelevant’. Table 3.4 shows Jenny’s feedback.

No.	E-resource name	Rank
1	Fuzzy logic	Irrelevant
2	Bayesian Networks	Very relevant

Table 3.4: Result step 5

Step 6:

### Chapter 3: Proposed architecture

The PLE application is able to predict e-resources that Jenny might prefer. Some of the e-resources predicted by the PLE application are shown in Table 3.5.

No.	E-resource name
1	A tutorial on learning with Bayesian networks
2	Modeling physiological processes with dynamic Bayesian networks
3	A causal mapping approach to constructing Bayesian networks
4	A driver fatigue recognition model based on information fusion and dynamic Bayesian networks
5	Dynamic Bayesian networks for information fusion with applications to human-computer interfaces
6	Bayesian networks: a practical guide to applications

Table 3.5: Result step 6

#### **Second scenario:**

This scenario consists of two parts.

- Part one: Predictive provider

Jenny expects the PLE application to make a prediction or recommendation for good sources for the type of e-resources she prefers. She has 5 preferences for such sources (providers) for her type of e-resources as follows:

1. Cost plays important part in selecting a provider for Jenny. She would rather only pay in the case of the provider giving her access to extremely good e-resources (i.e. IEEE and Ethos).
2. Jenny would prefer that the PLE application were able to change the provider during her period of study. For example, in the first year she might use the library to gain some background on a particular topic but in her second year she might look for conference papers; a good provider for those is IEEE. Thus, the PLE application should have the facility to change the provider for Jenny depending on her academic year. Accordingly, the provider in the first year will be different from the second and different again in the third year because her preferences will change over time.
3. Jenny is typically advised by her supervisor to look at specific providers such as Ethos, the library and IEEE. Therefore, she takes the advice of her supervisor as one of her preferences in selecting the providers.

### Chapter 3: Proposed architecture

4. In certain cases, she prefers to glean only general information about a specific topic. For example, if she wants to gain some background on context-aware systems, she might prefer to read a book on that topic; the best provider of books is the university library, with a probability of 85%.
5. Sometime she needs to see more details (read in depth) on a topic. If Jenny wants to learn more about “Modelling context-aware systems” for example, a good provider would be IEEE, with a probability of 90%.

#### **Part two: Predict computer science keyword (sub-subject)**

Let us assume that Jenny studies computer science (this is her main subject). Jenny’s preferences in computer science are: Z notation and UML, computer crime, digital evidence, dynamic systems and fuzzy logic. These are the general preferences for Jenny; however, at certain times she has specific preferences only, such as only UML. In her case, UML refers to requirement engineering, which in turn refers to the software engineering; this is the sub-subject of the computer science which Jenny must look for.

The second case is that she has two preferences, for example, Z notation and computer crime. Z notation refers to formal methods, which in turn refers to software engineering. However, computer crime refers to the forensic computing, which in turn refers to computer security. As a result, there are two sub-subjects for this situation: software engineering and computer security. Jenny wants the PLE application to provide her with a probability for each sub-subject (software engineering and computer security). Thus, for example, the PLE application might predict that the probability for looking for software engineering is 75%, whereas the probability for looking for computer security is 25%.

This second scenario (both parts) will be implemented in Chapter 4.

## Chapter 3: Proposed architecture

### **3.8. Summary**

This chapter has provided details of the context-aware PLE architecture. It began with introducing the requirements and the motivation. Then, the proposed approach, called “context-aware PLE architecture”, was presented in detail. Subsequently, the chapter presented a walk-through of the proposed architecture. In addition, the advantages and the limitations of the proposed architecture were detailed. Following this, implementation considerations were presented. Finally, this chapter ended by providing a case study for the proposed architecture.

In the next chapter, the implementation of the Predication components will be presented.

## **Chapter 4: Implementation of the predication component**

### Objectives:

- To introduce artificial intelligence techniques
- To present the fundamental principles of Bayesian networks
- To determine the objective of the implementation
- To present first DBN model
- To present second DBN model
- To present the steps for building the DBN



### **4.1. Artificial Intelligence**

A system that is able to reason is called an Artificial Intelligence (AI) system. AI has the ability to draw conclusions based on the information gathered from different sensors. The features of AI have been summarised in [60, 61, 104], and some of these are: flexibility, adaptability, learnability, reasoning and dynamism.

To model a context-aware system when the context is ‘uncertain’, an AI technique must be used. The AI technique must have a property that fuses the sensed information in order to resolve any conflict and thus increase the level of confidence in the results.

There are various widely deployed AI techniques, including fuzzy logic, neural networks, Probabilistic Logic and Bayesian networks, and these are introduced below:

- Neural networks are able to fuse the outputs of multiple sensors. A neural network has the ability to learn associations between input and output; the input and output are important factors for an observer [62, 63, 64]. The learning capability of a neural network is very high but its internal state is of no interest to the observer; it can be described as a black-box for the user. Neural networks are good for applications that demand control, estimation and system identification [65]. However, the limitation of neural networks is their lack of design techniques. Neural network techniques are not suitable for working in a dynamic sensor configuration environment, as each sensor requires a unique input node and each possible sensor-set configuration requests that it is specifically trained. In addition, training a neural network is normally a slow process [63].
- Fuzzy logic is defined by (Michael) in [60] thus: “Fuzzy logic is determined as a set of mathematical principles for knowledge representation based on degrees of membership rather than on crisp membership of classical binary logic” [60]. Fuzzy logic has been applied to various applications such as robotics and medicine. It is used to represent vague concepts and is also used for reasoning [66]; each component described in fuzzy logic has a membership degree of

## Chapter 4: Implementation of the predication component

either 1 or 0. A serious weakness with this is that, because fuzzy logic uses approximations rather than fixed and exact data, it is not a good candidate for managing systems that require extreme precision [67, 68].

- Probabilistic Logic

The aim of the probabilistic logic is to facilitate logical assertions that are associated with a probability [60]. This technique makes it possible to make a statement, such as “the probability of B is less than  $\frac{1}{2}$ ” and “the probability of A is at least twice the probability of B”; where B and A are random variables. Furthermore, using probabilistic logic allows us to write rules to reason about an event’s probability of occurring relative to other events. These rules can be used for resolving conflicts between context information obtained from different sources. A serious weakness with this tool, however, is that it does not offer adequate expressive rules to capture the uncertainties and dependencies between variables, or to model the temporal aspects of the domain [69].

- Bayesian Networks

A Bayesian network (BN) is a graphical model resulting from a marriage between probability theory and graph theory, and provides a natural tool for dealing with uncertainty, knowledge representation and inference [70, 71]. A Bayesian network is defined as having each variable represented by a node in a graph [72]. The direct dependencies between the variables are represented by directed edges. The directed edges between two variables in a BN represent a causal relationship. The relationship between two variables is specified by the conditional probabilities table (CPT). BNs are precise and efficient in representing and storing conditional probabilities [66, 72, 73].

For this research, a BN has been chosen to implement the Prediction component in the proposed architecture for the following reasons:

- It is able to model time series data, taking into account the previous and current context of the user [73, 74].

## Chapter 4: Implementation of the predication component

- A BN is more likely to be used when reliable statistical data have been gathered [73]. For this reason, it is proposed to carry out a survey in order to gather the reliable data.
- BNs have been used successfully in many domains including medical systems and aeroplane technology.
- BNs are efficient at combining uncertain contextual information from a wide range of sensors to deduce high level contextual information [68].
- BNs offer a hierarchical framework to systematically represent information from diverse modalities at different levels of abstraction and systematically account for uncertainties [75].
- BNs are able to handle incomplete data and can model causal relationships between variables [76].

### **4.2. Bayesian networks**

A BN is a suitable tool as it has the ability to present different sensors (each collecting uncertain information) and connect them all into one system; it also has the ability to present the level of uncertainty [60, 72, 77]. A BN is more likely to be used when reliable statistical data have been gathered. In addition, a BN is based on probability theory, which is the best technique for dealing with random variables or uncertain knowledge. Furthermore, a BN works very well with forecasting where statistical data are available, and it can explain how it arrives at a particular solution. It is able to compute the probability for any variable, when other variables are known [73, 75].

Before describing BNs in depth, it is useful to remind the reader that BNs are also known by other names such as Bayesian belief networks, belief networks and recursive graphical models [78]. This study will use the term Bayesian network (BN). A BN can be described as an expert system because it has ability to learn from its experience. There are many good introductions to BNs, including [73, 79, 80]. This chapter will briefly present the main concepts for understanding BNs.

## Chapter 4: Implementation of the predication component

### **4.2.1. Static Bayesian network**

A static Bayesian network (SBN) has been defined in [73, 80, 81, 82, 83, 84]; it represents each variable with a node on a graph. The direct dependencies between the variables are represented by directed edges. The directed edges between any two variables within a BN represent a causal relationship. Furthermore, an SBN is a graphical model that consists of variable and edges. An SBN is able to model one point in time, which means that the arcs in the SBN do not say anything about time. Later in this chapter, a demonstration how an SBN can be developed will be shown.

Each variable has states and each state has a value or probability. Each variable can have two states, for example, the variable ‘Smoker’ has two states, which are true and false. The value of each state may be true = 80%, false = 20%. In additional, a variable can have more than two states. For example, the variable ‘Temperature’ has three states: high, medium and low. The value for each state may be high = 60%, medium = 30% and low 20%.

In general, building a BN involves four steps [63, 61], and these are:

- The first step is to identify the variables and their states.
- The second is to draw the relationship between the variables in the network.
- The third is to parameterise the network by determining the Conditional Probability Table (CPT) associated with each node.
- The final step is to perform the inference algorithm.

Bayesian inference mechanisms are based on Bayes’ theorem. In the mechanisms, the probability of the state in a node can be inferred, when the evidence for another node is given. BN theory is based on Bayes’ Theorem, as presented in Equation 4.1 below [85, 86, 71].

$$p(h/e) = \frac{p(e/h)p(h)}{p(e)} \quad (4.1)$$

where:

$p(h/e)$  is the probability of the hypothesis (h), given the input (e).

## Chapter 4: Implementation of the predication component

$p(e/h)$  is the probability of the input (e), given the hypothesis (h).

$p(h)$  is the prior probability of the hypothesis (h).

$p(e)$  is the prior probability of the input (e).

### **4.2.2. Pattern recognition**

When the uncertain information is available, pattern recognition should be used. This is a basic human brain function [87], and is how the brain recognises an object. Pattern recognition confirms certain patterns and distinguishes other patterns. It has several processes or steps, which are: sensing, analysis, decision and result. Pattern recognition techniques simulate the human brain and can produce uncertain results about the domain. Based on pattern recognition, various approaches have been studied, and one of them is Bayesian networks.

### **4.2.3. BN topology (structure)**

The first step in working with a BN is to identify variables and to group them into sets of jointly exclusive events in order to form the target variables (hypotheses node). This entails classifying the observable data such that they reveal something meaningful about the variables, and then clustering them into ‘information variables’ [73, 88].

Once the variables are identified, the nodes should be ordered. A node represents a variable. It is axiomatic that the network structure depends on the ordering of the nodes. Suppose that there are five variables, which are: cancer (C), pollution (P), dyspnoea (D), x-ray (X) and smoker (S) [73]. There are many possible ways of ordering the nodes; in this case, the following order was chosen:

$$\langle p, s, c, x, d \rangle \quad (4.2)$$

Now the network structure can be built. In adding the variable  $p$ , it can be said that  $p$  is the root node. The next variable is  $S$ ; by adding  $S$  the network should look like Figure 4.1:

## Chapter 4: Implementation of the predication component



Figure 4.1: BN structure, step 1

Before continuing, a question should be asked here. Is  $S$  independent of  $p$ ? The answer is yes, because there is no relationship between these two variables.

Now, let us add the next variable, which is  $c$ . When adding  $c$ , it must be realised that there is a relationship between  $c$  and  $p$ , and also between  $c$  and  $s$ . Thus, arcs should be drawn from  $p$  and  $s$  to  $c$ . Accordingly, the network structure looks like Figure 4.2.

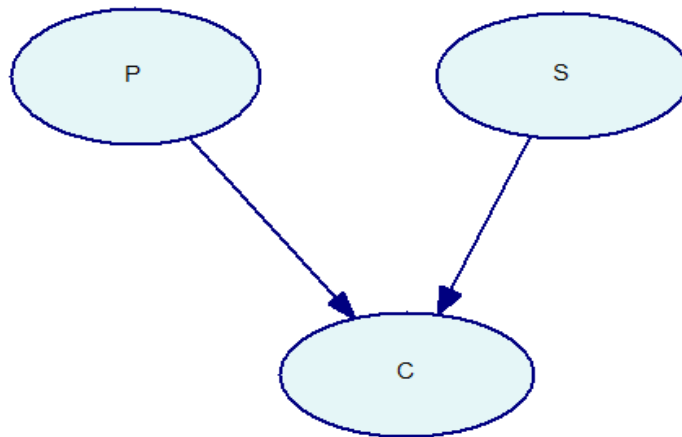


Figure 4.2: BN structure, step 2

Finally, the last two variables,  $d$  and  $x$ , are added. It has been noted that  $c$  (cancer) is the main cause for these two variables ( $d$  and  $x$ ), i.e. there is a relationship between  $c$ ,  $d$  and  $x$ . Accordingly, an arc is drawn from  $c$  to  $d$ , and also from  $c$  to  $x$ . The network is now completed and is shown in Figure 4.3.

## Chapter 4: Implementation of the predication component

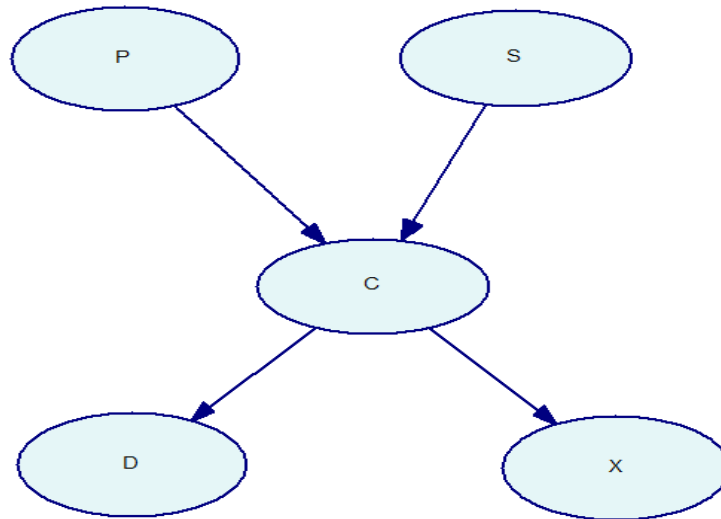


Figure 4.3: BN structure, step 3

Thus, the best way to order the nodes is to add the root cause first, and then the variables that are influenced by it directly. As a result, the model will tend to be more compact and the network will be much easier to deal with. Furthermore, the model will use less computer memory [73, 85].

### **4.2.4. Conditional Probability Tables (CPT)**

When the BN topology (structure) has been built, the next step is to measure the relationships between the connected nodes. This measurement can be done by specifying a conditional probability table (CPT) for each node [80, 88]. In each node, all the possible combinations of states of the parent nodes have been located. Each combination is called an instantiation of the parent set. The size of the CPT depends on both the number of possible states of a node and the number of parents the node may have [73]. The strength of the relationship between variables is measured by the conditional probability table associated with each node.

A BN can be modelled into levels of quality and quantity [89]. The quality level represents the directed acyclic graph, where nodes represent variables. In addition, the directed arcs describe conditional independence relationships. For example, Figure 4.4 presents the quality level of the BN.

## Chapter 4: Implementation of the predication component

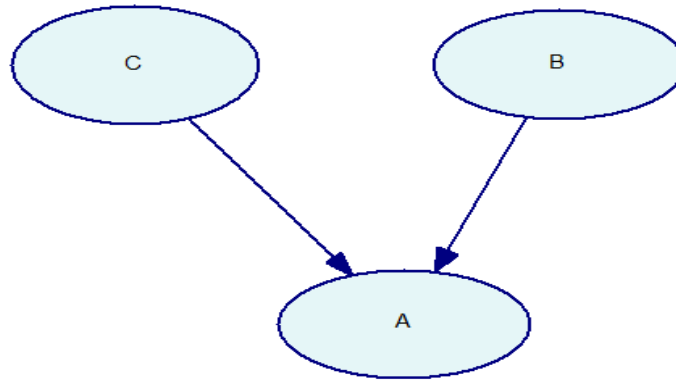


Figure 4.4: Quality level

The quantity level represents the CPT, where each variable has a conditional probability table; each variable  $X$  has a set of possible states [90]. The quantity level in a BN describes a set of parameters that represent the strength of the dependencies.

In order to understand what a CPT looks like, an example is here introduced. The probability table of node  $A$  reflects the probability for all possible combinations of  $A$ ,  $B$  and  $C$ . There are two possible states for node  $B$ , which are  $B1$  and  $B2$ , and two possible states for node  $C$ , which are  $C1$  and  $C2$ . Finally, node  $A$  has two possible states,  $A1$  and  $A2$ . Figure 4.5 has more details.



## Chapter 4: Implementation of the predication component

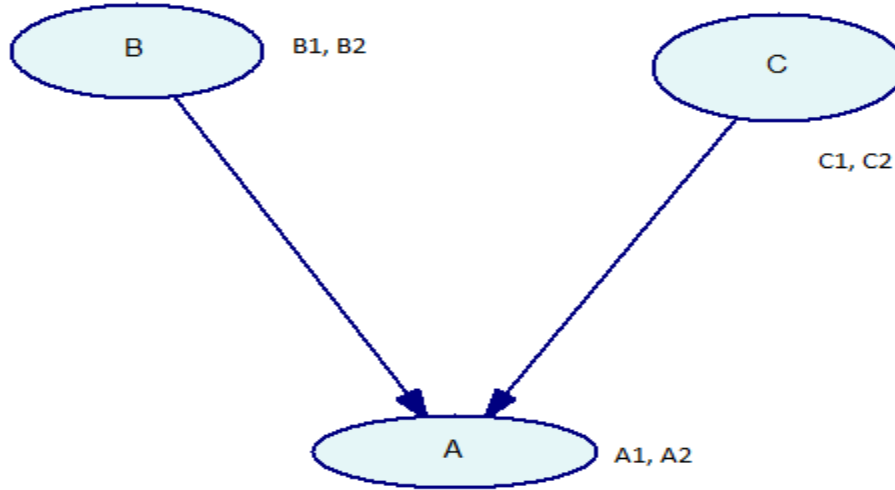


Figure 4. 5: Quantity level

The combination of all the possible states of the three nodes consists of  $2^3 = 8$  elements. The CPT of the combination is shown in Table 4.1. A major problem with the CPT is that of size; the size of a CPT grows with the number of parents and the number of states of each variable. It is recommended that the size of a CPT be kept to a minimum [73], and so a CPT for node A has been created. The probabilities within the CPT are then estimated. Clearly, from Table 4.1, the probability of A1 can be read from the CPT: if  $B = B1$  and  $C = C1$ , then  $P(A1/B = B1, C = C1) = 0.3$ .

C	C1		C2	
B	B1	B2	B1	B2
A1	0.3	0.3	0.6	0.4
A2	0.7	0.7	0.4	0.6

Table 4. 1: CPT for node A

Parameterisation is another name for specifying the conditional probability table (CPT) for each variable. The parameters are used to estimate future behaviour. According to [80, 91], a parameter can be estimated. The probabilities of each state can be obtained from statistical information [92, 93].

## Chapter 4: Implementation of the predication component

There are two different approaches to obtain the values (probabilities) of the CPT for each state of the node in the network [59, 60, 61, 62]. These approaches are described below:

1. First approach: Obtain the probabilities from different resources. The probability should be obtained manually by critically analysing different resources or papers. This also includes studying the relationship between the nodes.
2. Second approach: Collecting the probabilities from trained data by carrying out a survey. A survey must be designed to suit a specific system. After collecting data from the survey respondents, the data file is imported into the Bayesian network software; this will calculate (parameterise) the probabilities for each states of a node.

This research uses the second approach because there are no previous studies that offer the required information. Thus, a survey has been designed and is presented in the Appendix. The survey comprises 14 questions. The targets were research students in the field of technology in the UK, and 180 respondents participated in the data collection process. After collecting the data from the respondents, the data file was imported into GeNIe software. GeNIe's 'learn parameters' feature was used to calculate the probabilities for each variable. In order to reduce costs and control the time spent conducting this survey, it was distributed to students via e-mail.

### **4.2.5. Conditional independence**

To understand how a BN works, one should consider the relationship between the Bayesian network structure and conditional independence. There are three types of relationship in BNs; they are: causal chains, common causes and common effects. These relationships are discussed in more detail below [72, 73]:

- **Causal chains**

The following example explain the understanding of the causal chains[73]. Whether a patient has dyspnoea depends directly on whether or not he/she has cancer, and whether

## Chapter 4: Implementation of the predication component

the patient has cancer depends on whether or not he/she is a smoker, as shown in Figure 4.6.

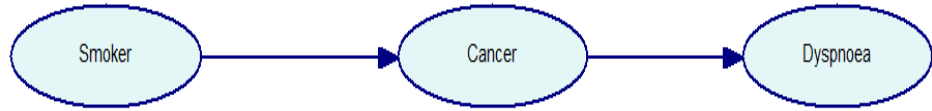


Figure 4. 6: Causal chains

If a doctor does not know whether his patient has cancer, but the doctor knows that this patient is a smoker, the doctor's belief that his/her patient has cancer will increase. This indicates that the patient may suffer from dyspnoea. On the other hand, if the doctor knows that the patient has cancer, the doctor does not need to know whether or not the patient is a smoker; this can be written thus:

$$p(d/s^c) = p(d/c) \quad (4.3)$$

The probability that a patient has dyspnoea, given that the patient is a smoker and has cancer, is same as the probability of dyspnoea, given that the patient has cancer. That is, dyspnoea is conditionally independent of the patient being a smoker, given that the patient has cancer.

- **Common causes**

Knowing that a patient has cancer affects the likelihood of two things happening, which are x-ray and dyspnoea [73]. In this case, cancer is a common cause for both x-ray and dyspnoea. If a doctor knows that the patient has dyspnoea, this will increase the probability that the patient has cancer, which in turn affects the likelihood that the patient will have an x-ray, as shown in Figure 4.7.

## Chapter 4: Implementation of the predication component

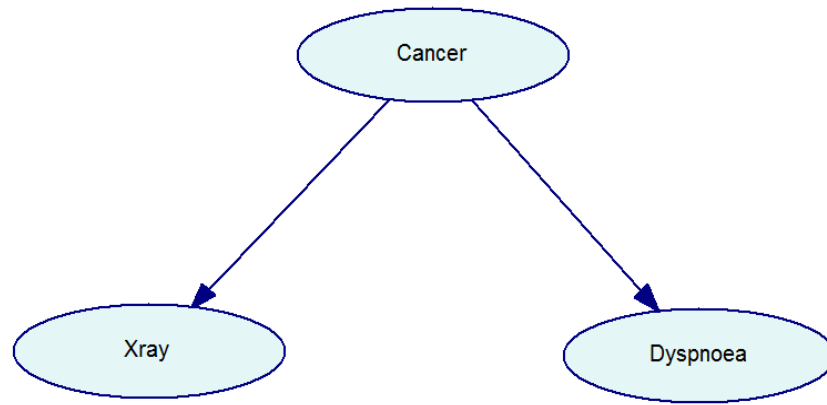


Figure 4. 7:Common causes

The probability of having an x-ray, given both dyspnoea and cancer is same as the probability of having an x-ray, given cancer.

$$p(x/d^c) = p(x/c) \quad (4.4)$$

This means that knowing one symptom (for example, dyspnoea) will affect the likelihood of having cancer, which in turn affects the likelihood of the other symptom (for example, having an x-ray). The common cause is conditionally independent, in the same way as causal chains. Conditional independence means that the x-ray node is only affected by the cancer node.

- **Common effects**

There are two causes for cancer, which are smoker and pollution [73]. If a doctor knows (observes) that a patient does not live in a polluted area, and then the probability of the doctor's belief in the patient being a smoker will increase. Thus, the two causes (smoker, pollution) are common effects which are conditionally dependent on each other. Figure 4.8 below represents common effect in a BN.

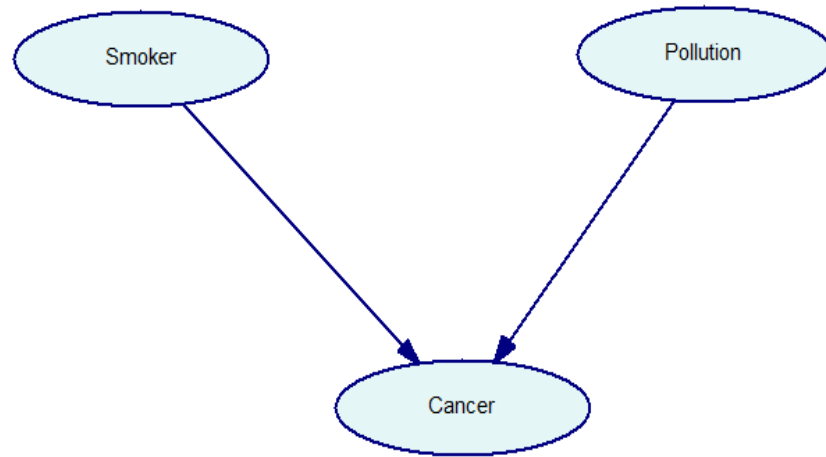


Figure 4. 8: Common effects

### **4.2.6. Inference in a Bayesian network**

This section explains how reasoning functions in a BN. When the state of a certain variable is observed, the BN can compare the current condition with the new information [94]. This process is called belief updating, inference or probability propagation; in this research, the word inference will be used. A significant feature of a BN is that it has the ability to carry out inference; it is the task of calculating the probabilities of each state of each variable, when the states of the other variables are known, i.e. inference is the process of computing the probability of the state of the variable  $x$ . [81].

A BN is also able to calculate new belief, once new information has been sensed or observed but that sensed information must be specified [95]. For example, if a doctor knows that his patient lives in a polluted area, this means that  $p$  (pollution) = True. However, in some cases, not all input can be observed, and this is where a BN is very useful, as it can deal with missing data.

There are various algorithms for inference in BNs, and choosing an algorithm depends on the network structure. Thus, a software engineer must choose an algorithm that fits the system. According to [73], it is possible to build a BN without fully comprehending how inference algorithms work, as inference algorithms are already implemented in all

## Chapter 4: Implementation of the predication component

BN software. There are two main types of inference: exact inference and approximate inference [81].

Exact inference algorithms include polytree and clustering inference. A BN is called a polytree if there is one path between any two nodes in the network. A polytree is a directed acyclic graph, and every polytree is a multi-tree. In the polytree network, a single node must have a unique path to every other node; in other words, there are no loops in the network. Polytree inference is an efficient inference technique which is used when the network is a polytree. For example, Figure 4.9 shows a BN wherein there is one path only between node N1 and N5.

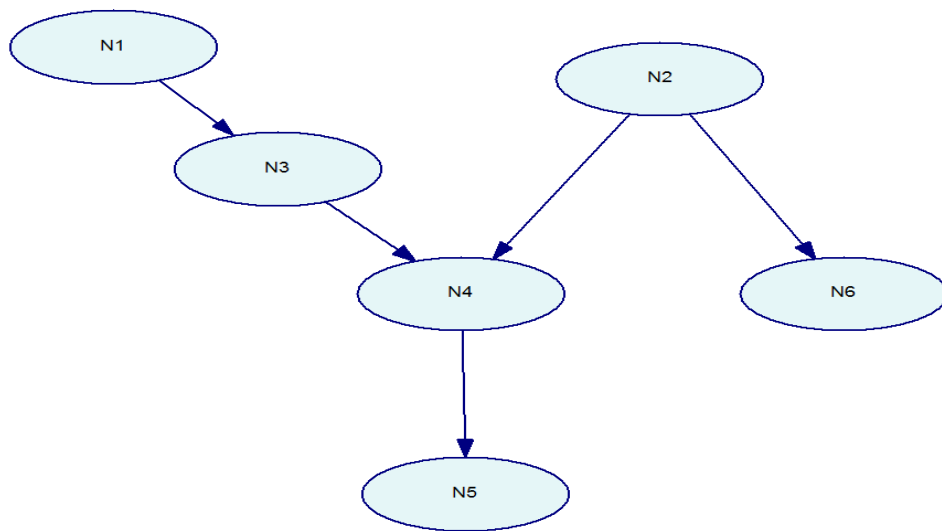


Figure 4. 9: One path

Clustering inference is used when there are two nodes connected via at least two paths. This type of network is called a “multiply-connected network”, as shown in Figure 4.10. Such a network must use clustering inference.

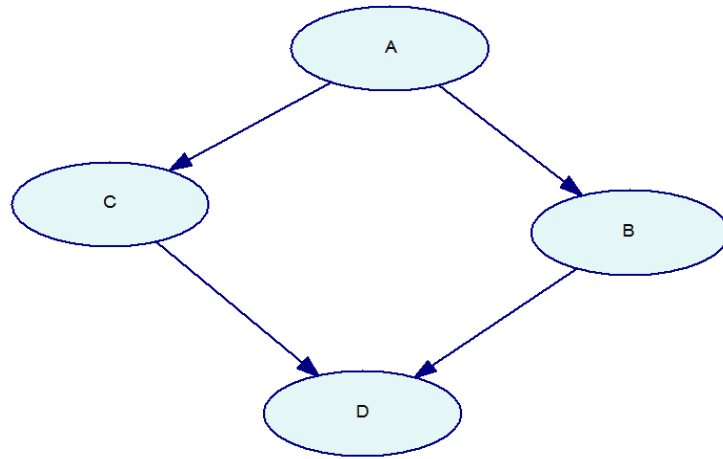


Figure 4. 10: Two paths

There are two stages to clustering inference. In the first stage, multiple connections should be removed, so that the network becomes a polytree. The second stage is to perform polytree inference on the network.

Exact inference is usually used with small networks. However, when a BN is large, approximate inference has to be used [73]. There are many approaches for approximate inference that have been developed, including: logic sampling, likelihood weighting and Markov chain Monte Carlo. For more information about inference algorithms, there is a good survey presented in [81].

### **4.2.7. Dynamic Bayesian networks (DBN)**

This section discusses the concept of dynamic Bayesian networks (DBN). BNs can be classified into static Bayesian networks (SBNs) and dynamic Bayesian networks (DBNs). Static BNs (SBNs) work with a single time instant and are particularly suitable when modelling systems that not evolve over time, which can represent a limitation to their use [75, 73].

As a result, a special structure, i.e. a DBN, must be designed to address temporal issues in BNs. According to [96, 97], DBNs have the ability to deal with many inputs and outputs, and they can use the past (history) to predict the future. DBNs have been used successfully in many areas.

## Chapter 4: Implementation of the predication component

In most Artificial Intelligence techniques, it is important to identify any time series. This is because a system can evolve over time [97, 98, 99]. A DBN can be as described as single BN modelling different points in time. In a DBN, at any point in time, the system should depend on the previous point in time. This means that a system at time  $t$  will depend on its previous point  $t-1$  [72], as shown in Figure 4.11.

Parovan was the first to introduce a time series that represents a BN at different times [73]. When there are  $N$  BN models for different times, i.e.  $BN1, BN2, \dots, BNn$ , an arc that connects these models represents the time series (this is actually how DBNs originated). Thus, the function of a DBN is to model change over time. A DBN model is therefore expected to change dynamically, and each result is used as feedback to improve the final result.

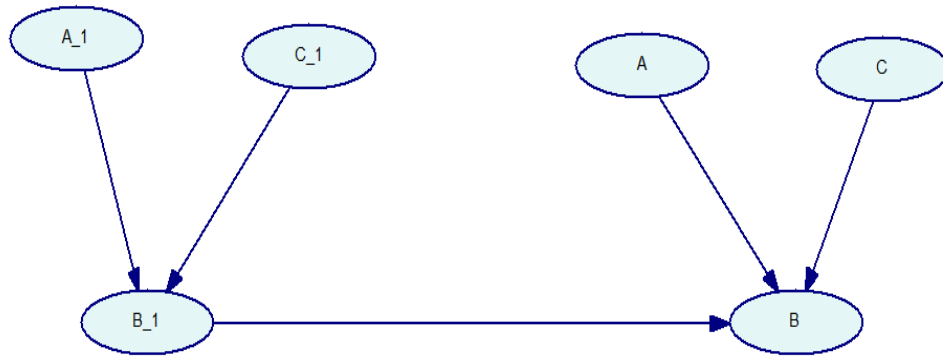


Figure 4. 11: DBN

### **4.2.8. Bayesian network software**

In this section, some major software tools, which can be used for modelling BNs, are introduced. These software tools have been built to support research into BNs; some are free of charge and others are commercially available. Some of these tools are: Bayesian Lab, MATLAB, Bayes Net Toolbox and GeNIe [73, 100, 101]. GeNIe is used for this research, for the following reasons [73, 102]:

- It has the ability to support both BN and DBN.



## Chapter 4: Implementation of the predication component

- It includes a polytree inference algorithm, which is used in this research. It offers exact inference in polytree, which is suitable for the DBN modelling proposed in this research.
- It is freely available for download.
- Its software is fast and well documented.
- It provides a Graphical User Interface (GUI).
- It is able to offer bar charts for the nodes and it displays the probabilities of each state graphically.
- It allows its users to specify any node of interest (hypothesis node).

The model presented in this work has been created using GeNIe software. GeNIe was developed at the Decision System Laboratory of the University of Pittsburgh, and has been successfully implemented in research and industry. Users of GeNIe must be familiar with the basic idea of BNs [103] but there is no need for users to know how the inference technique works. Most of GeNIe's users are using it as a research tool [103], and it now has thousands of users, including independent researchers, university colleges and various branches of industry. One of its features is that it allows users to add comments to each part of the network, including nodes, arcs and node states. Finally, this tool can be run under Windows and Linux [62].

GeNIe is implemented in Visual C++. Models developed using GeNIe can be inserted into any application. Figure 4.12 shows a screen shot taken from GeNIe2.0; it provides a user-friendly interface. The GeNIe tutorials in [101] are well structured, allowing the user to learn how to use it steadily, without overpowering the user with too much detail. It is suggested that the user proceed through the tutorials in the order in which they are shown.

## Chapter 4: Implementation of the predication component

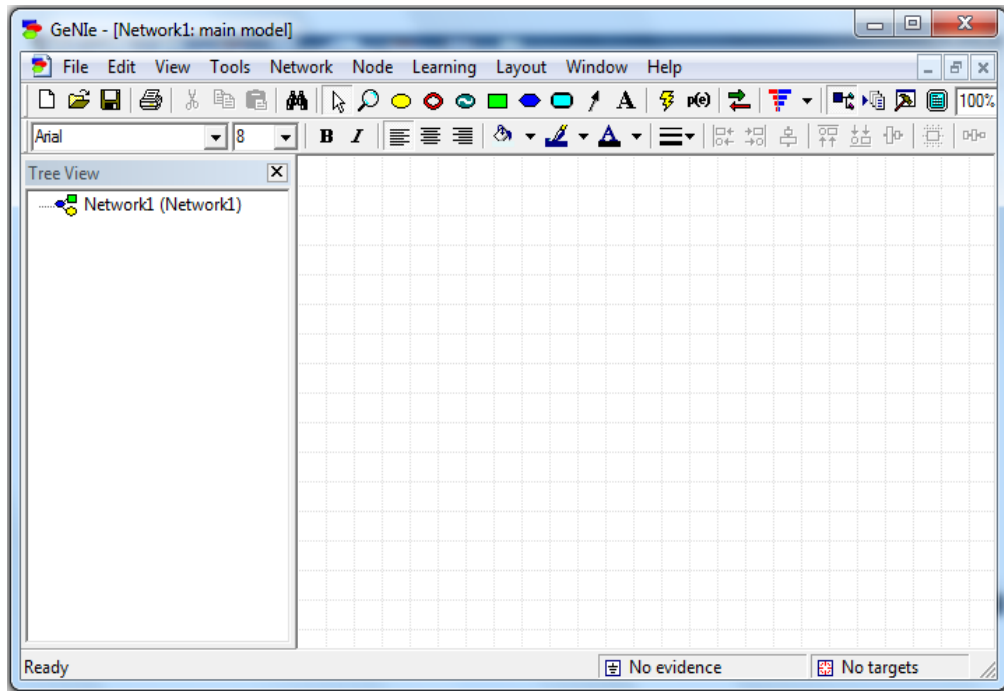


Figure 4. 12: GeNIe user interface

### 4.3. Overview of the implementation

This section considers implementing the Prediction component within Generic subsystem. The Prediction component is able to learn, reason and be dynamic. This component has the ability to predict user context. The description of the implementation in this chapter is based on the case study which is presented in Chapter 3. The Prediction component is able to receive user context from Profile Detector.

The AI technique known as DBNs is used to implement the Prediction component in the proposed architecture. The result is a DBN model that is able to obtain information from various sensors. DBN modelling was chosen because it supports the following features:

- It can deal with inaccurate information.
- It has the ability to model time series data which evolves over time.
- It is able to combine prior data with current data.

## Chapter 4: Implementation of the predication component

Effectively, a DBN is a set of static Bayesian networks which are connected by sequential time slices. In this study, a first order DBN is used. This means that the hypothesis node at time slice ( $t$ ) depends on the variables at time slice ( $t$ ) and the hypothesis node at time slice ( $t-1$ ). In contrast, different kind of sensors can be used to gather contextual information about the user's behaviour by performing reasoning under uncertainty using DBN. Two DBN models are presented in this chapter. The first DBN model is used to predict a suitable provider for the user. The second model presented uses keywords from the Computer Science vocabulary to predict a suitable subject for the user.

In both models, this research has tried to minimise both the number of nodes and the states of those nodes in the network. This is because the computations can become high complex if the node has many parents or if it has a large number of states. For example, if there are three nodes (one child and two parents), the CPT for the child node will be  $2^3 = 8$ . Adding one more node (parent node), the CPT for the child node will be  $2^4 = 16$ . Thus, the CPT will be highly complicated. The details for building these DBNs are presented in the next section.

### **4.4. First DBN model**

This section presents first DBN model, which is used to predict a suitable provider for the user. Four steps will be presented to create the first model:

- The first step is to identify the variables in the DBN network and their states.
- The second step is to draw the network graph.
- The third step is to parameterise the network.
- The final step is to perform the inference algorithm.

#### **4.4.1. Defining the network variables (nodes).**

Based on the case study in Chapter 3(second scenario, first part), a user needs a PLE application that suggests good sources (provider) for the type of e-resources he or she need. Some variable that affects users to choose suite provider are presented in the case

## Chapter 4: Implementation of the predication component

study in Chapter 3. In addition, a survey is distributed to identify other variable as it is shown the appendix. The next section presents more details about these variables.

The hypothesis node in this network is the provider node, which represents several providers. The variables (which affect the hypothesis node) have been divided into three variables. These variables are: supervisor advice, type of material and membership. These variables are taken as key contributors to the hypothesis node, which in this case is the provider. All these variables are discussed below:

**Variable 1: Supervisor advice.** Students are typically advised by their supervisor to look at specific providers such as Ethos, library and IEEE. Therefore, supervisor's advice is one of the contextual features that affect a student when selecting their provider.

**Variable 2: Type of material.** Type of material is very important to students. A student may look for a thesis, book or conference paper. Each type of material has its own provider. For example, if the student is looking for a thesis and they have reached the writing up period, they may wish to look at how the thesis is structured. In this case a good provider to use is Ethos. Or if a student is looking for books, a suitable provider to use is the library. Finally, when the student is looking for a conference paper, a suitable provider is IEEE. As a result, knowledge of the type of material can assist the system to recognise which provides the user needs.

Types of material are influenced by three variables: **requiring details, general information** and the **year of study**. If a student is in the third year of their work, it is more likely that they will be looking for a thesis, because they are writing up their thesis; whereas if a student is in the first year of their research they are more likely to be reading books to obtain the fundamental principles relating to their topic. Finally, if the researcher is looking for more detail, they should look for a conference paper as this will provide information about their topic.

**Variable 3: Membership.** Membership also plays an important role in the choice of provider. If the student has an account in Ethos or goes to the library regularly, he is a member of these providers. There is one variable which might influence users in terms

## Chapter 4: Implementation of the predication component

of membership of a provider: this is the **cost** of the provider. If a student needs a free cost provider, they might be a member of the library. On the other hand, a student may be able to use a provider which is not free, for example Ethos. It is clear that membership can be taken as a key contributor, affecting the choice of provider.

The hypothesis and the variables have been determined. Each variable has several states. Tables 4.2 show the variables and their possible states. Whereas Table 4.3 shows the hypothesis node with its states.

<b>Variables</b>	<b>State1</b>	<b>State2</b>	<b>State3</b>
Supervisor's advice	Ethos	Library	IEEE
Type of material	Thesis	Book	Conference Paper
Year of study	First	Second	Third
General	Yes	No	
Details	Yes	No	
Membership	Ethos	Library	
Free Cost	Yes	No	

Table 4. 2: The possible states of the variables

<b>Hypothesis node</b>	<b>State1</b>	<b>State2</b>	<b>State3</b>
Provider	Ethos	Library	IEEE

Table 4. 3: The states of the hypothesis node

### **4.4.2. Drawing the Network graph**

The second step when designing a BN is specifying the causal relationship between the variables. This step requires a network graph be drawn. In order to produce a DBN, it is necessary to fuse the available variables into one model. The network graph is drawn in terms of a directed acyclic graph (DAG). The first DBN proposed model is shown in Figure 4.13. The hypothesis node in the proposed model is the provider, where the variable node above the hypothesis node indicates the information node. The model is

## Chapter 4: Implementation of the predication component

unrolled for two time slices. The hypothesis node at time slice ( $t$ ) depend on the variables at time slice ( $t$ ) and the hypothesis node at time slice ( $t-1$ ).

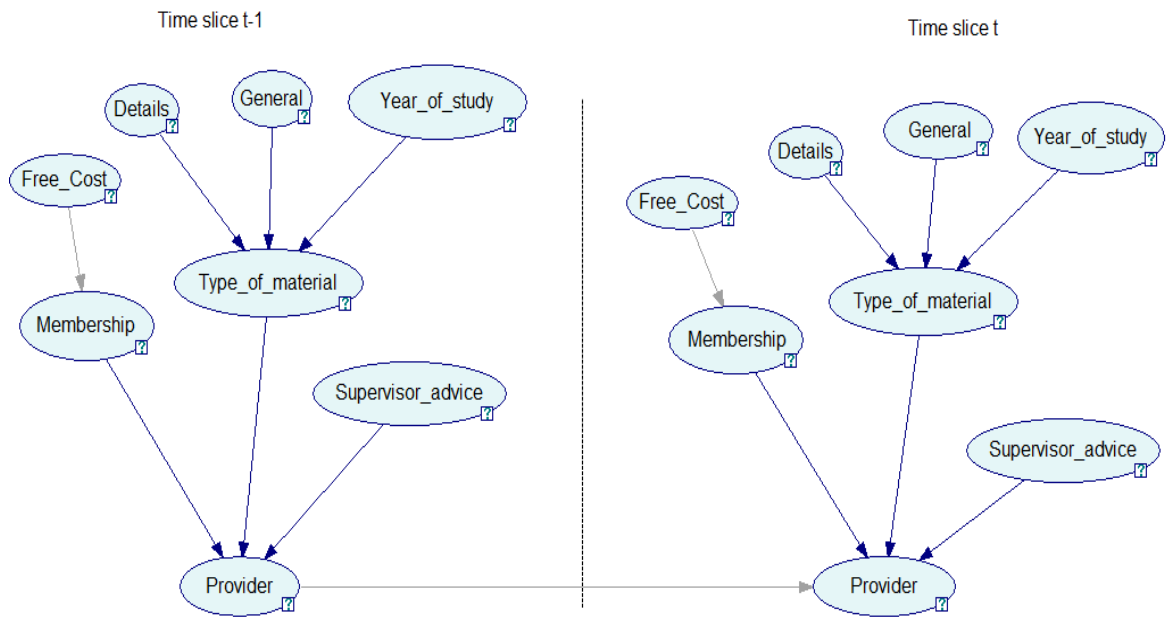


Figure 4. 13: DBN provider model

### 4.4.3. Parameterising the first DBN model

This section presents the parameterisation of the first DBN model. After the DBN structure has been created, as shown in the previous section, the third step is to parameterise the network. The value (probability) of each state must be selected for each node in the network. This means that the conditional probability for each of the connected nodes in the network is calculated. As mentioned in chapter 4, all the probabilities(values) for the CPT were calculated using the data collected from the Survey. The data file was imported to GeNIe. GeNIe's 'learn parameters' feature was used to calculate the probabilities for each variable. Table 4.4- 4.7 show the CPT for the node in the network.

Parent node			Probability of Provider node		
	Supervisor		Ethos	Library	IEEE

## Chapter 4: Implementation of the predication component

Membership	advice	Type of material			
Ethos	Ethos	Thesis	0.98997078	0.007014611	0.003014609
		Book	0.7	0.29	0.01
		Conference paper	0.75	0.05	0.2
	Library	Thesis	0.65	0.18	0.17
		Book	0.9859375	0.01328125	0.00078125
		Conference paper	0.35	0.3	0.35
	IEEE	Thesis	0.79	0.01	0.2
		Book	0.0086111111	0.98166667	0.0097222222
		Conference paper	0.010032653	0.058791837	0.93117551
Library	Ethos	Thesis	0.65	0.3	0.05
		Book	0.0012755102	0.53938776	0.45933673
		Conference paper	0.38	0.3	0.32
	Library	Thesis	0.32	0.52	0.16
		Book	0.010004	0.938792	0.051204
		Conference paper	0.14	0.63	0.23
	IEEE	Thesis	0.47863946	0.26068027	0.26068027
		Book	0.43612457	0.12754325	0.43633218
		Conference paper	0.11	0.82	0.07

Table 4. 4: CPT for provider node

Parent node			Probability of type of material node		
Details	General	Year of study	Thesis	Book	Conference paper
Yes	Yes	First	0.00030864198	0.99830247	0.0013888889
		Second	0.000125	0.9985	0.001375
		Third	0.66652893	0.33295455	0.00051652893
	No	First	0.05	0.05	0.9
		Second	0.38478935	0.10005829	0.51515236

## Chapter 4: Implementation of the predication component

		Third	0.35277551	0.14694694	0.50027755
No	Yes	First	0.15997694	0.75001922	0.09000384
		Second	0.13	0.79	0.08
		Third	0.99486111	0.0045833333	0.00055555556
	No	First	0.67	0.17	0.16
		Second	0.88	0.06	0.06
		Third	0.75	0.1	0.15

Table 4. 5: CPT for type of material node

Parent node	Probability of membership node	
Free Cost node	Ethos	Library
Yes	0.24	0.76
No	0.88	0.12

Table 4. 6:CPT for membership node

Nodes	States	Probability
Cost	Yes	0.54444772
	No	0.45555228
Details	Yes	0.66110772
	No	0.33889228
General	Yes	0.65555325
	No	0.34444675



## Chapter 4: Implementation of the predication component

Year of study	First	0.37221971
	Second	0.26111505
	Third	0.36666524
Supervisor advice	Ethos	0.27222002
	Library	0.31111077
	IEEE	0.41666921

Table 4. 7: CPT for other nodes

### **4.4.4. Inference the first DBN model.**

The final step in building the DBN is performing inference on the hypothesis node, which here is the provider node. As discussed earlier in Chapter 4, there are two inference algorithms in BN: for exact and approximate inference. This research uses exact inference because the first DBN model that is proposed is small and uncomplicated. Exact inference includes polytree inference, clustering inference and other types. Polytree inference is suitable for the DBN model, since there is one unique path between each pair of nodes. Consequently, polytree inference has been applied, as shown in Figure 4.14. When applying inference to the hypothesis node at the current time slice, the hypothesis node at the previous time slice is considered as one of the information variables. The result of the inference will be presented in chapter 5 as part of the simulation of the DBN model.

## Chapter 4: Implementation of the predication component

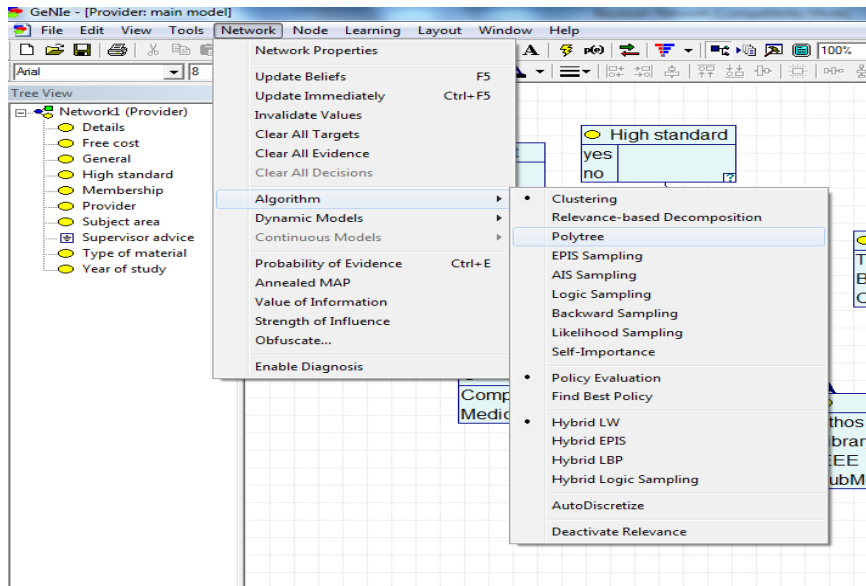


Figure 4.14: Polytree inference

### 4.5. Second DBN model

The second DBN model presented in this chapter is used to predict the most suitable computer science keywords for the user. This implementation based on the case study, which is presented Chapter 3. The steps for building this model are same as for the previous model.

#### 4.5.1. Defining the network variables (node)

The user needs a PLE application that suggests or predicts a suitable subject for the user. Some variables that affect user to identify good keyword are gathered by the survey (the survey is presented in appendix) and other variable are already identified in the case study which presented in Chapter 3(second scenario, second part).

The hypothesis node in this network is the computer science keyword node that represents several keywords for computer science. The hypothesis node not observes directly. However, it can be measured by knowing other subject that affects the computer science keyword. The variables (which affect the hypothesis node) have been divided into three variables, which are artificial intelligence, computer security, and software engineering. Details for these variables will be shown in next section.

## Chapter 4: Implementation of the predication component

**Variable 1: Artificial intelligence:** Artificial intelligence is a computer science subject and is a technology and a branch of computer science that studies and develops intelligent machines and software. There are two variables that affect the artificial intelligence variable. These are fuzzy logic and dynamic system.

**Variable 2: Computer security:** Computer security aims to protect information and services from unintended or unauthorised access. Computer security is also a very important variable that affects the choice of computer science keyword. The computer security variable is affected by two other variables which are digital evidence and computer crime.

**Variable 3: Software engineering:** Finally, software engineering plays an important role for a computer science keyword. Software engineering aims to design, develop and maintain software. The software engineering variable is affected by two variables which are UML and Z notation.

The hypothesis node and the variables node have been determined. Each variable has several states which refer to the state that the variable can take. Tables 4.8 shows the variables and their possible states. Table 4.9 shows the hypothesis node with its states.

Variables	State1	State2
Z notation	Yes	No
UML	Yes	No
Computer crime	Yes	No
Digital evidence	Yes	No
Dynamic system	Yes	No
Fuzzy logic	Yes,	No
Software engineer	Formal Methods	Requirements Engineering
Computer security	Forensic Computing	Security Management
Artificial Intelligence	Computational Modelling	Mobile Robotics

Table 4. 8 The states of the variables

## Chapter 4: Implementation of the predication component

Hypothesis node	State1	State2	State3
Computer science keyword	Software engineer	Computer security	Artificial Intelligence

Table 4. 9: The states of the hypothesis node

### 4.5.2. Drawing the network graph

After identifying the network variables, the next step is to draw the network graph. In order to produce a DBN, it is necessary to fuse the available variables into one system. The network graph is drawn in terms of a directed acyclic graph (DAG). The second DBN proposed model is shown in Figure 4.15. The hypothesis node in the proposed model is the computer science keyword, where the variable node above the hypothesis node is the information node. The hypothesis node at time slice ( $t-1$ ) is treated as one of the information variable. The model is unrolled for two time slices. The hypothesis node at time slice ( $t$ ) depends on the variables at time slice ( $t$ ) and the hypothesis node at time slice ( $t-1$ ).

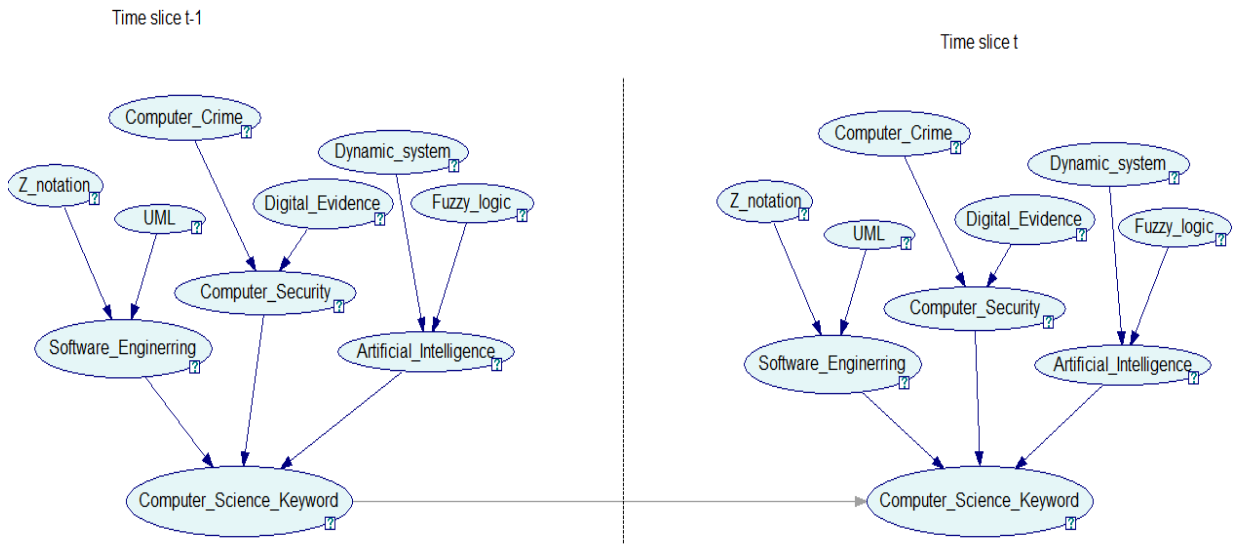


Figure 4. 15: DBN computer science keyword model

## Chapter 4: Implementation of the predication component

### **4.5.3. Parameterising the second DBN model**

The section presents the parameterisation of the second DBN model. The second model presents uses keyword from the computer science vocabulary. The third step is to parameterise the network. The value (probability) of each state must be selected for each node in the network. This means that the conditional probability for each of the connected nodes in the network is calculated. As mentioned in chapter 4, all the probabilities(values) for the CPT were calculated using the data collected from the Survey. The data file was imported to GeNIe. GeNIe's 'learn parameters' feature was used to calculate the probabilities for each variable. Table 4.10 -4.14 show the CPT for the node in the network.

<b>Child node</b>		<b>Probability of Artificial intelligence node</b>	
Fuzzy logic	Dynamic system	Computational Modelling	Mobile Robotics
Yes	Yes	0.035609228	0.96439077
	No	0.0066666667	0.99333333
No	Yes	0.071183673	0.92881633
	No	0.70774316	0.29225684

Table 4. 10: CPT for artificial intelligence node

<b>Child node</b>		<b>Probability of Computer Security node</b>	
Digital evidence	Computer crime	Forensic Computing	Security Management
Yes	Yes	0.024	0.976
	No	0.00076124567	0.99923875
No	Yes	0.62839587	0.37160413
	No	0.78207994	0.21792006

Table 4. 11: CPT for computer security node

## Chapter 4: Implementation of the predication component

Child node		Probability of software engineering node	
UML	Z notation	Formal Methods	Requirement Engineering
Yes	Yes	0.010288269	0.98971173
	No	0.14126874	0.85873126
No	Yes	0.56752674	0.43247326
	No	0.44429667	0.55570333

Table 4. 12: CPT for software engineering node

Parent nodes			Probability of Computer Science Keyword node		
Computer Security	Software Engineering	Artificial Intelligence	Artificial intelligence	Computer Security	Software Engineering
Forensic Computing	Formal Methods	Computational Modelling	0.032706897	0.086896552	0.88039655
		Mobile Robotics	0.95380367	0.036018362	0.010177966
	Requirement Engineering	Computational Modelling	0.0065583673	0.0072946939	0.98614694
		Mobile Robotics	0.24493269	0.033846154	0.72122115
Security Management	Formal Methods	Computational Modelling	0.0012398457	0.60863142	0.39012873
		Mobile Robotics	0.012625	0.039	0.948375
	Requirement Engineering	Computational Modelling	0.019424603	0.97495196	0.0056234336
		Mobile Robotics	0.00085561497	0.20144385	0.79770053

Table 4. 13:CPT for the hypothesis node.

## Chapter 4: Implementation of the predication component

<b>Nodes</b>	<b>States</b>	<b>Probability</b>
Z notation	Yes	0.39170825
	No	0.60829175
UML	Yes	0.30120571
	No	0.69879429
Computer crime	Yes	0.34420906
	No	0.65579094
Digital evidence	Yes	0.21133976
	No	0.78866024
Dynamic system	Yes	0.29341717
	No	0.70658283
Fuzzy logic	Yes	0.22146806
	No	0.77853194

Table 4. 14: CPT for other nodes

### **4.5.4. Inference the second DBN model**

This section presents the inference of the second DBN model. Similar to previous model, this model uses exact inference because the DBN model that is proposed is very small and uncomplicated, and once again polytree inference is suitable for the first DBN model, since there is one unique path between each pair of nodes. Consequently, polytree inference has been applied to this model.

## **4.6. Summary**

The first part in this chapter has discussed the fundamental principles of Bayesian networks. The chapter began by introducing Artificial Intelligence techniques; the advantages and the limitations of each were presented. An illustration of the main reasons for choosing a Bayesian network was then illustrated.

Following this, a detailed description of Bayesian networks was given, including the definition of a BN, BN topology and conditional probabilities. Moreover conditional independence, inference and dynamic BNs were described.

## Chapter 4: Implementation of the predication component

This chapter presented the implementation of the Prediction component within Generic subsystem. The Prediction component is able to learn reason and be dynamic. This component has the ability to predict user context. The description of the implementation in this chapter is based on the case study, which is presented in Chapter 3.

The AI technique known as DBNs used to implement the Prediction component in the proposed architecture. The result was a DBN model that is able to obtain information from various sensors. A DBN is a set of static Bayesian networks, which are connected by sequential time slices. In this study, first order DBNs used. This means that the hypothesis node at time slice ( $t$ ) depends on the variables at time slice ( $t$ ) and the hypothesis node at time slice ( $t-1$ ). In contrast, it gathers information from different kinds of sensors regarding the user behaviour by performing reasoning under uncertainty using DBN. Two DBN models proposed in this chapter. The first DBN model used to predict a suitable provider for the user. The second model presented uses keywords from the Computer Science vocabulary to predict a suitable subject for the user.

In the following chapter, the evaluation and the simulation are presented in this research.



## **Chapter 5: Evaluation and Simulation**

Objectives:

- To present an evaluation of the proposed architecture
- To simulate the proposed DBN models

## Chapter 5: Evaluation and Simulation

### **5.1. Introduction**

This chapter presents an evaluation of the proposed architecture by offering a comparative study regarding the PLE architecture, assessing it against the current PLE architectures. This involved comparing the proposed approach with four other PLE architectures. Also, the chapter presents the simulation of the DBN models. The simulation in this research is that to checks that the proposed models satisfy their intended purposes.

### **5.2. Evaluation of the proposed architecture**

The evaluation in this study demands a comparative study with other PLE architectures in order to identify the strengths and the weaknesses of the proposed architecture. The evaluation in this research is structured as follows: firstly, an overview of the available PLE architectures is presented. Secondly, Criteria for assessment are detailed. Thirdly, a comparison study is presented to explore the strengths and weaknesses of each PLE architecture.

#### **5.2.1. Available PLE architectures**

This section presents the currently PLE architectures. The field of PLE has a short history and therefore, a few PLE architectures are proposed in a literature (these architectures were reviewed in Chapter 2). Table 5.1 below displays general information about the existing PLE architectures including the purposed approach in this research. The general information includes the name of each PLE architecture, the developer's name, and the date.

No.	PLE name	Developer name	Date
1	PLE based on free web services	Zubrinic	2008
2	PLE model	Wilson	2009
3	Personal Learning Environment Framework (PLEF)	Chatti	2010
4	PLE Framework	Hongyu	2010
5	Context-aware PLE	Mafawez	2013

Table 5. 1: General information on existing PLE architectures

## Chapter 5: Evaluation and Simulation

### **5.2.2. Criteria for assessment and critical analysis**

PLE's are still in the process of evolving; consequently, there are no descriptors for standard critical points, making PLE systems difficult to evaluate comparatively. Therefore, criteria for assessment for this evaluation, based on the features of the existing PLEs, together with the features of context-aware PLE architecture. These features are used as criteria for assessment to enable to evaluate the proposed architecture. The following section presents all these features:

1. **User feedback:** This feature enables the user to give feedback on how they have benefited from the system. The aim of this is to attempt and comprehend the user's perspective and opinions, and thence to anticipate the user's desires and needs. User feedback is based upon a ranking received directly from the system's users; the ranking ranges from irrelevant to relevant. The outcome of this feature may then be used to contribute to the user recommendations development phase. These feedback data should indicate how useful the e-resource is to users. For example, after a student has evaluated the information (e.g. read a conference paper), he/she will then be in a position to rank the usefulness of the information from 'irrelevant' to 'relevant'.
2. **Context history:** Context history is a significant aspect of deliberation when describing the history of the user. A system must use context history for future processing. Context history has the ability to store high-level information that has been previously delivered to the user. This storage is essential to prevent duplicate information from being sent to the user at a later point in time. For example, a research student does not want to receive e-resource more than one.
3. **Learner control:** There is one major practical limitation in VLEs: the institutions or education systems maintain control over the resources. On the other hand, the users of the PLE system are able to monitor and control their own learning; the system is independent, which gives the user complete control over managing the system. Thus, the ownership of PLE has shifted from the institution to the learner.
4. **Sharing information:** The PLE system allows the user to share information with a third party if necessary. For example, learners will be able to share their

## Chapter 5: Evaluation and Simulation

e-resource with others. This feature allows users to communicate with other users who are interested in the same subject, and during such communication, the user can collect and store additional information.

5. **Personalisation:** The users of the PLE system have the option to personalise tools based on interest. For example, when a user wishes to deal with photographs, a suitable application (e.g. Flickr) can be used. Another example would be YouTube, which can be used to display a wide variety of user-generated video content. Therefore, different tools can be personalised based on the user's interests.
6. **Artificial Intelligence:** PLE architecture can be implemented using Artificial Intelligence (AI) techniques, which facilitate the fusing of sensory information in order to resolve conflicts, and to increase confidence levels. AI has the ability to draw conclusions based on the information gathered from different sensors. PLE system has to predict or suggest new user contexts based on existing context information stored in the context history, together with information stored in the user profile and user feedback
7. **Lifelong learning:** A PLE should support lifelong learning, which means that it should be available to learners throughout their lives. In other words, it refers to continuously allowing learners to update their knowledge. This means that in order to obtain knowledge, learning is no longer required to be located in a specific place and time. In addition, there is no specific age for learning, and so it is available to people of all ages. PLE system is able to support long-term activities, thereby allowing the learner access to e-information anytime and anywhere.
8. **User profile:** PLE architecture must include a user profile, allowing the user to access his or her profile in order to create, add and delete information via the user interface. The user can provide information relating to his or her various interests. It is anticipated that the user's profile will change over time and that the user will be able to modify it directly. User profiles have different lifetimes. For example, a user may be interested in using it for a short period of time, which will then naturally time out and be removed. In contrast, a user may have a lifelong interest in a certain subject, and as that interest evolves, it will remain

## Chapter 5: Evaluation and Simulation

in the profile until deleted by the user (by the user only). In addition, the user profile accepts suggestions about changing preferences. The user profile is affected by the user's reaction to information gathering; for example when a user deletes information, the generic part of the system can send an alert or adapt to the user profile in order to create a dynamic phase that will provide better results in the future.

- 9. User location:** The use of user location has been increasing in context-aware systems. Advertisements can be sent to the user's device based on its location [107], i.e. consumers receive advertisements based on their location. Most context-aware advertisements are delivered to mobile devices using Short Message Services (SMS).

### **5.4.3 Discussion and result**

The final results of this comparative study are presented Table 5.2. It can be seen from the table that the best PLE architecture is the context-aware PLE, which has 8 out of the 9 features; it is missing just one feature, which is user location.

There are no significant differences between the PLEF and PLE models, as both of them have 4 out of 9 features. These features are: learner control, sharing information, personalisation and lifelong learning. This shows that both architectures are fairly weak, as they do not contain 5 out of the 9 features. This indicates that the proposed approach in this research is stronger because it contains the 4 that both of these architectures have, but it also has five additional features.

PLE based on web services has 3 out of the 9 features. These features are: learner control, sharing information and lifelong learning. This shows that this approach is very weak, as it lacks 6 out of the 9 features. This architectural approach is thus poorer than the approach developed in this research.

The weakest PLE architecture is PLE Frameworks. This architecture lacks 7 out of the 9 features. The architecture supports only two features, which are information sharing and learner control. This approach is thus extremely poor, and is of little worth if compared with the proposed approach in this research.

## Chapter 5: Evaluation and Simulation

No.	PLE architecture  Features	Context- aware PLE	PLE based on web services [9]	PLEF [26]	PLE model [25]	PLE Framework [8]
1.	User feedback	√	x	x	x	x
2.	Context history	√	x	x	x	x
3.	Learner control	√	√	√	√	√
4.	Sharing information	√	√	√	√	√
5.	Personalisation	√	x	√	√	x
6.	Artificial Intelligence	√	x	x	x	x
7.	Lifelong learning	√	√	√	√	x
8.	User profile	√	x	x	x	x
9.	User location	x	x	x	x	x
10.	Total features	9	9	9	9	9
11.	Total available features	8	3	4	4	2
12.	Total missing features	1	6	5	5	7

Table5. 2: Summary of comparisons

## Chapter 5: Evaluation and Simulation

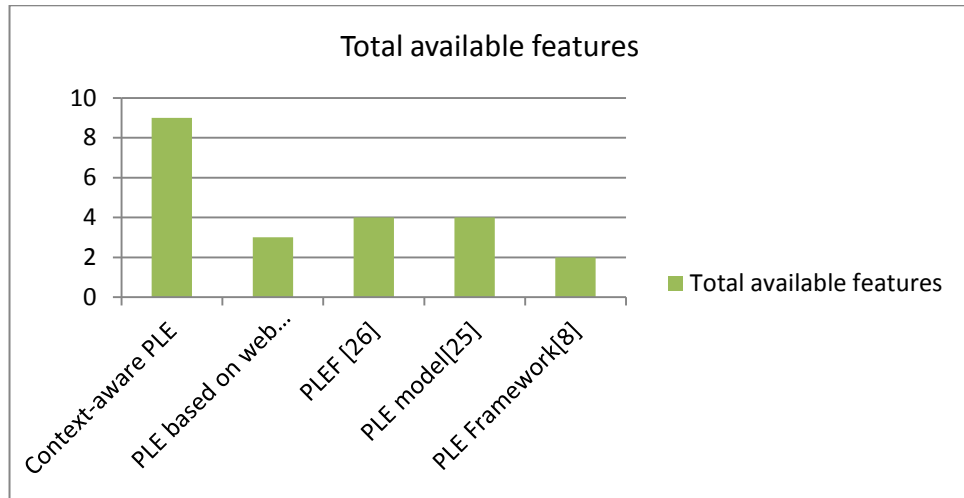


Figure 5.1: Summary of comparisons

Figure 5.1 compares all the PLE architectures. There are 9 features in total; however, no architectural approach contains all 9 features. In summarising all the comparisons, it is apparent that the context-aware PLE approach is by far the strongest, as it contains 8 out of the 9 features, and is still under development.

### 5.3. Simulation of the DBN proposed models

#### 5.3.1. Introduction

This part presents the simulation of the DBN models presented in Chapter 4. The simulation in this research is that to checks that the proposed models satisfy their intended purposes”. There are two DBN models to validate; the first is used to predict a suitable provider for the user, and the second to predict a suitable subject for the user.

The methodology for the simulation consists of three steps:

- Select time step count
- Select the states of the nodes
- Apply inference

## Chapter 5: Evaluation and Simulation

### 5.3.2. Simulation of the first model

This section focuses on the simulation of the first DBN model. The main purpose of this model is to predict a suitable provider for the user, e.g. IEEE, ACM. The simulation in this section emphasises the accuracy of the model. The following subsections explain the simulation process of first model.

#### Step 1: Select the time step count

Time step count means how many times the user uses the model. The time step count must be selected during the simulation process. DBN is a model that evolves over time, and for this reason, this research uses two time step counts in order to include both the previous and the current states of the nodes. Figure 5.2 shows the GeNIe user interface that allows the user to select the time step count.

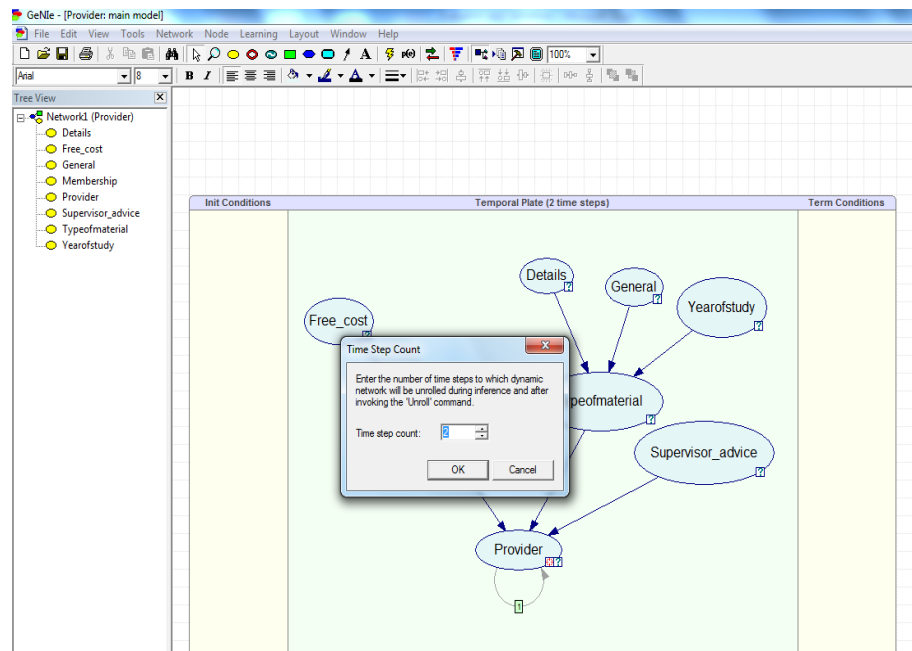


Figure 5. 2: Time step count

#### Step 2: Select the states of the nodes

The second step in the simulation process is to select the states of the nodes. Figure 5.3 shows how the states of each node can be selected in GeNIe (Figure 3.13 in Chapter 3 shows an interface for PLE applications to obtain the states of the nodes).



## Chapter 5: Evaluation and Simulation

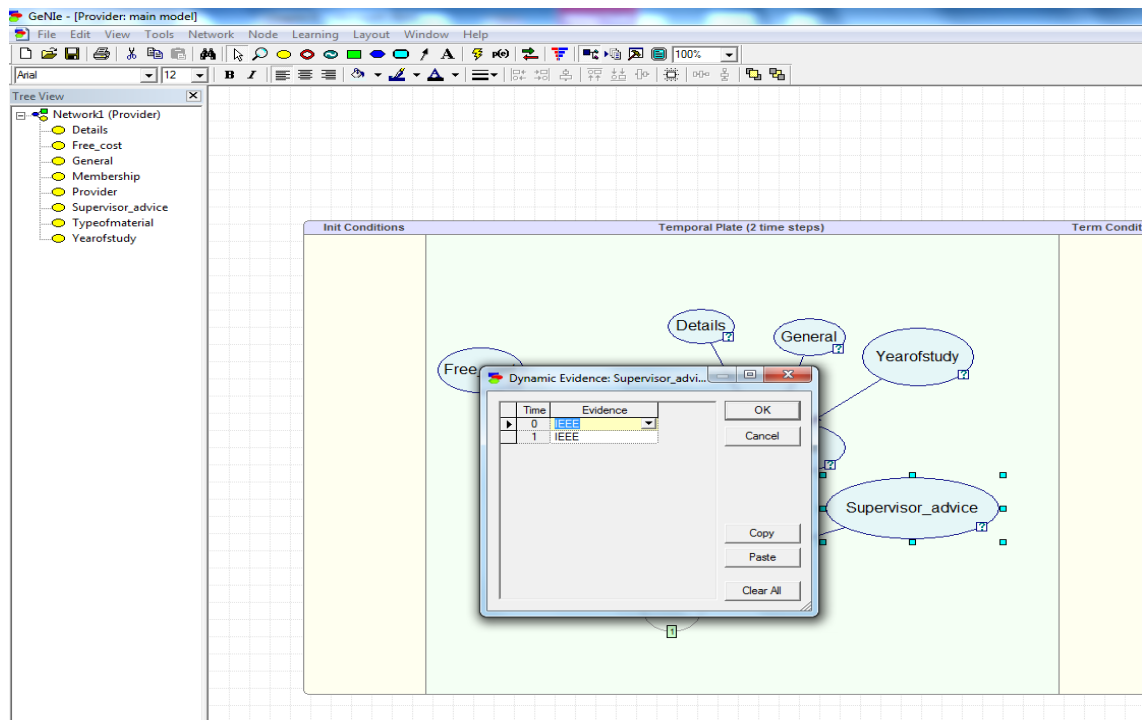


Figure 5. 3: Select the states of the node in GeNIe software

One of the features of a DBN model is that it allows the user to insert its inputs from different nodes (input node). The first model uses multiple nodes as an input for the model. The input nodes in this model are as follows: supervisor advice node, year of study node, general node, details node and free-cost node. The user should select the state for each input node. Table 5.3 shows some possible combinations of the states of the input nodes. This research assumes that in two different times, the states of a node stay the same.

The following abbreviations are used for simplicity: supervisor advice= SA, year of study=YFS, general=G, details=D, free cost=FC, IEEE=IE, Ethos= ETH, library=LI, first= FI, second=SE, third=TH, yes= Y, no=N.

## Chapter 5: Evaluation and Simulation

	<b>SA</b>	<b>YFS</b>	<b>G</b>	<b>D</b>	<b>FC</b>
1.	IE	SE	N	Y	N
2.	IE	TH	N	Y	N
3.	IE	TH	N	Y	Y
4.	IE	SE	Y	Y	Y
5.	ETH	FI	Y	N	Y
6.	LI	FI	Y	N	Y
7.	LI	FI	Y	N	Y
8.	LI	FI	Y	Y	Y
9.	LI	SE	Y	Y	Y
10.	ETH	TH	Y	Y	Y
11.	IE	TH	N	Y	N
12.	IE	TH	N	Y	N
13.	ETH	TH	Y	Y	N
14.	ETH	TH	Y	N	N
15.	ETH	SE	N	Y	N
16.	IE	FI	Y	N	N
17.	LI	FI	Y	N	Y
18.	IE	FI	Y	N	Y

Table 5. 3: Combinations of the states of the input nodes

### **Step 3: Apply inference**

This section presents the inference step for the first DBN model. Table 6.13 illustrates the inference results on the provider node and on the probability of its state.

The inference type used in this model is the Polytree algorithm. The algorithm produces a result based on two inputs, which are: the result from the survey and knowing the state of the input node selected by the user.

It is clear that from Table 5.4 that the probability for the provider node reaches the highest level when its state is Library (number 8 in the Table); the probability to use the library = 0.93824826 (as shown in Figure 5.4), and this is because all the states of the input nodes are interested in library. The user is advised by their supervisor to use library and the student is in his/her first year. In addition, the user is interested in

## Chapter 5: Evaluation and Simulation

general information. Furthermore, he/she is looking for a cost-free provider (library). The probability presented in the table proves the validity of the first DBN model.

On the other hand, the probability of the provider node attains the lowest level when its state is Ethos; the probability to use Ethos = 0.032021328 (as shown in Figure 5.5). This is because all the states of the input nodes show that the user is no longer interested in using Ethos. The user has been advised by the supervisor to use the library and he/she is in his/her first year, i.e. the user has not reached the stage where he/she needs to know how to structure a thesis. He/she is also looking for more general knowledge, for example books. Furthermore he/she is looking for a cost-free provider (Ethos is not free).

No.	SA	YFS	G	D	FC	Provider	Probability
1.	IE	SE	N	Y	N	IE	0.65655912
2.	IE	TH	N	Y	N	IE	0.5576712
3.	IE	TH	N	Y	Y	IE	0.68077584
4.	IE	SE	Y	Y	Y	IE	0.42829274
5.	ETH	FI	Y	N	Y	IE	0.36934936
6.	LI	FI	Y	N	Y	IE	0.0759778
7.	LI	FI	Y	N	Y	LI	0.89007951
8.	LI	FI	Y	Y	Y	LI	0.93824826
9.	LI	SE	Y	Y	Y	LI	0.92835258
10.	ETH	TH	Y	Y	Y	LI	0.38101965
11.	IE	TH	N	Y	N	LI	0.1601482
12.	ETH	TH	N	Y	N	LI	0.067573422
13.	ETH	TH	Y	Y	N	ETH	0.89828026
14.	ETH	SE	Y	N	N	ETH	0.74197196
15.	ETH	SE	N	Y	N	ETH	0.84449391
16.	IE	FI	Y	N	N	ETH	0.16927264
17.	LI	FI	Y	N	Y	ETH	0.032021328
18.	IE	FI	Y	N	Y	ETH	0.41122697

Table 5. 4: Inference for first model

## Chapter 5: Evaluation and Simulation

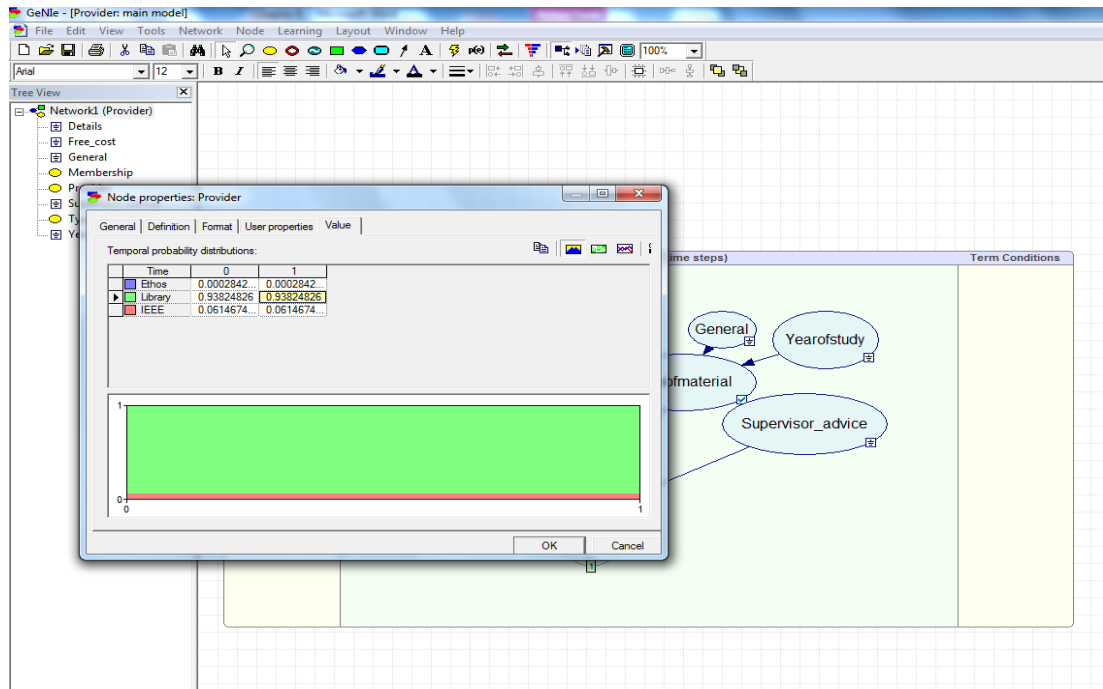


Figure 5. 4: Highest level of provider's state

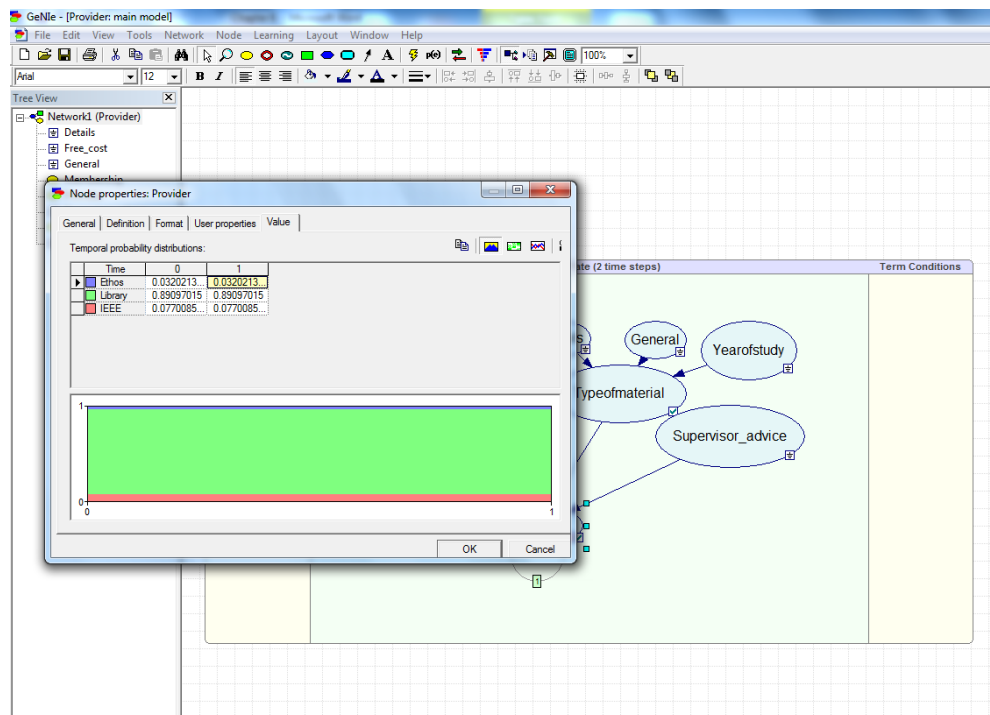


Figure 5. 5: lowest level of provider's state

## Chapter 5: Evaluation and Simulation

### **5.3.3. Simulation of the second model**

This section considers the simulation of the second DBN model. The main purpose of this model is that it uses keywords from the Computer Science vocabulary to predict a suitable subject for the user. The methodology for this model is the same as for the second one. The following subsections explain the simulation process of the first model.

#### **Step 1: Select the time step count**

The second model is similar to the first; the simulation in this model uses two time step counts in order to include both the previous and the current node states. This is because the model has the ability to evolve over time.

#### **Step 2: Select the states of the nodes**

The second step in the simulation process is to select the states of the nodes (Figure 3.14 in Chapter 3 shows an interface for PLE applications to the status of nodes). One of the features of the second DBN model is that it allows the user to obtain input from different nodes (input node). Therefore, the second model uses multiple nodes as an input for the model. The input nodes in this model are as follows: Z notation node, UML node, computer crime node, digital evidence node, dynamic system node and fuzzy logic node.

The state for each input node should be selected by the user. Table 5.5 shows some possible combinations of the states of the input nodes. This research assumes that in two different times, the states of the nodes stay the same.

The following abbreviations are used for simplicity: Computer Science keyword = CSK, Yes = Y and No = N, fuzzy logic = FL, dynamic system = DS, Z notation = ZN, UML = UML, computer crime = CC, digital evidence = DE, SE=software engineering, CS=computer security, AI= artificial intelligent.

## Chapter 5: Evaluation and Simulation

No.	ZN node	UML node	CC node	DE node	DS node	FL node
1.	Y	Y	N	N	N	N
2.	Y	Y	Y	N	N	N
3.	Y	N	N	N	N	N
4.	N	Y	Y	N	N	N
5.	N	N	Y	Y	N	N
6.	N	N	N	N	Y	Y
7.	N	N	Y	Y	N	N
8.	N	N	N	Y	N	N
9.	N	N	Y	N	N	N
10.	Y	Y	N	N	N	N
11.	N	N	N	N	Y	Y
12.	N	Y	N	N	Y	N
13.	N	N	N	N	Y	Y
14.	Y	N	N	N	Y	Y
15.	N	N	N	N	Y	N
16.	N	N	N	N	N	Y
17.	N	Y	N	N	N	N
18.	N	N	Y	N	N	N

Table 5. 5: combinations of the states of the input nodes

### **Step 3: Apply inference**

This section presents the inference process for the second DBN model. Table 6.15 illustrates the inference result on computer science keyword node, and on the probability of its state. The table below ensures the validity of the second DBN model.

It is clear that from Table 5.6 that the probability of the computer science node reaches its highest level when the state = software engineering. The probability reaches 0.76079846 (as shown in Figure 5.6). The main reason for this is that the input nodes are interested in software engineering. The state of both the Z notation node and the UML node are both = Yes, while the other states of the input nodes are = No; this increases the probability of the software engineering state to rise.

However, the probability of the computer security state for the computer science node is very low. The probability reaches 0.06091395 (as shown in Figure 5.7). The reason for this is that the input nodes are no longer interested in computer security. The states of

## Chapter 5: Evaluation and Simulation

the computer crime node and digital evidence node are = No, while the user might be interested in AI because the states of the dynamic system node and the fuzzy logic node are both = Yes.

No.	ZN	UML	CC	DE	DS	FL	CSK	Probability
1.	Y	Y	N	N	N	N	SE	0.76079846
2.	Y	Y	Y	N	N	N	SE	0.65852995
3.	Y	N	N	N	N	N	SE	0.67601078
4.	N	Y	Y	N	N	N	SE	0.65065434
5.	N	N	Y	Y	N	N	SE	0.38731847
6.	N	N	N	N	Y	Y	SE	0.5149331
7.	N	N	Y	Y	N	N	CS	0.59887331
8.	N	N	N	Y	N	N	CS	0.61217145
9.	N	N	Y	N	N	N	CS	0.25301392
10.	Y	Y	N	N	N	N	CS	0.17474248
11.	N	N	N	N	Y	Y	CS	0.06091395
12.	N	Y	N	N	Y	N	CS	0.06660867
13.	N	N	N	N	Y	Y	AI	0.42415295
14.	Y	N	N	N	Y	Y	AI	0.49041534
15.	N	N	N	N	Y	N	AI	0.40912237
16.	N	N	N	N	N	Y	AI	0.4363815
17.	N	Y	N	N	N	N	AI	0.087307795
18.	N	N	Y	N	N	N	AI	0.11455224

Table 5. 6: Inference for second model

## Chapter 5: Evaluation and Simulation

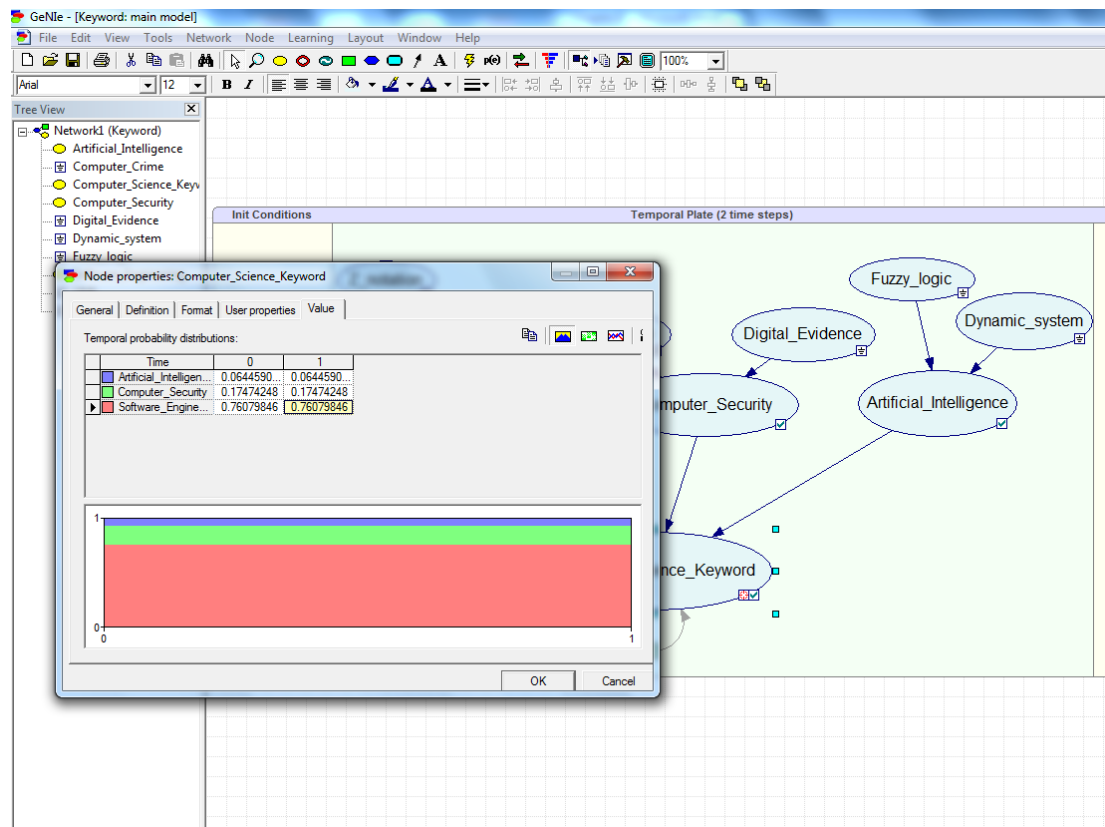


Figure 5. 6: Highest level of the computer science's state

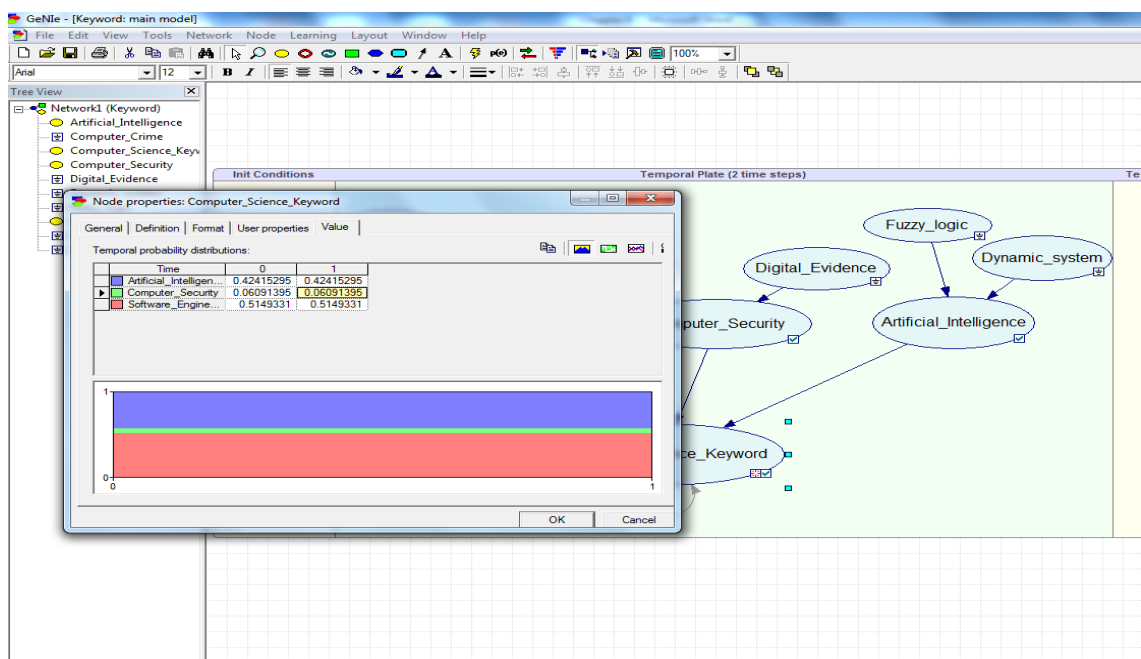


Figure 5. 7: lowest level the provider's state



## Chapter 5: Evaluation and Simulation

In summary, both DBN models are found to be successfully valid. The first DBN model is valid and fulfils its purpose, i.e. the model was able to predict a suitable provider for the user. In addition, the second model was able to predict suitable subjects for the user. During the simulation process, two time step counts were used in order to provide the previous and current states of the node. With more simulation, the models become more confident, and for this reason, 18 combinations were made in each model. Capturing more than one input node for each model has also given a more accurate result, and accordingly the models are becoming more confident.

### **5.4. Summary**

This chapter firstly evaluated the proposed architecture, assessing it against the current PLE architectures. This involved comparing the proposed approach with four other PLE architectures with respect to 9 functionality-based criteria, in order to demonstrate the strengths of the proposed architecture in this research. This evaluation revealed that the proposal has more features than the other PLE architectures, and the result is that the proposed approach has greater functionality and accordingly is better than the others.

Secondly, this chapter presented a simulation of the DBN proposed models. It demonstrated that the DBN models, which were presented in Chapter 4, are acceptable for use. The result of the simulation process shows that the DBN models fulfil their objectives. The simulation was further concerned with their level of accuracy. The findings of this chapter are that the DBN models are valid and that they can be used for real-world applications; the process also provided good simulation for the effectiveness of the DBN models.

## **Chapter 6: Conclusion and future work**

### **Objectives:**

- To summarise and conclude our findings
- To present the contributions of the research
- To revisit the success criteria
- To outline possible future work

## Chapter 6: Conclusion and future work

This research has successfully achieved the objectives proposed at the outset, and has provided an answer to the research question. This chapter summarises the research, clarifies the study's main contribution, revisits the success criteria, and state's the author's intentions for future work relating to this study.

### **6.1. Research summary**

In recent years, Virtual Learning Environments (VLE) has attained popularity in the context of higher education, due to their ability to provide flexible solutions for students and researchers. In response to the limitations of VLEs, a new generation of VLE – the Personal Learning Environment (PLE) – has been developed. As well as avoiding the limitations of VLEs, PLEs have additional features that allow students to control and develop new applications, such as Web 2.0 applications and social networking. Whilst PLEs, to date, have resolved some of the drawbacks of VLEs, there remains unexplored potential for PLEs to cover more areas. This research established as its premise that a context-aware system could cover the current limitations of PLE architecture. Context-aware systems are able to gather contextual information from a variety of sources without explicit user interaction and can adapt their operations accordingly. None of the existing PLE architectures have context-aware systems completely embedded with the architecture of PLE.

In this research, the developed approach “context-aware Personal Learning Environment architecture” is presented. The proposed architecture was built based on the concept of a context-aware system. The architecture has unique features. It can be described as a system that is driven by the user, based on user requirements. In addition, the architecture is dynamic, with the ability to continuously change and adapt to the user's preferences automatically. It is able to provide relevant e-resource that is important for the user. The PLE architecture is automatically responsive to each individual user on demand, and it is able to predict user preferences or interests. The automated refinement of user requests for e-resources enables results to improve over time.

## Chapter 6: Conclusion and future work

The implementation of the Prediction component in the proposed approach is presented using Bayesian Network techniques. The Prediction component within the proposed architecture has the ability to predict contextual information, to learn, to reason and to be dynamic. Two DBN models are presented in the implementation. The first is used to predict a suitable provider for the user. The second model presented uses keywords from the Computer Science vocabulary to predict a suitable subject for the user. The implementation was presented in Chapter 5 and the simulation for the implementation was presented as part of Chapter 6.

Chapter 6 presented an evaluation of the proposed architecture and the simulation of the two DBN models. Firstly, the chapter discussed the evaluation of the proposed architecture. This involved a comparison of our new approach with other PLE architectures, in order to highlight the strengths of our proposed architecture. Secondly, this chapter presented the simulation of the DBN models, demonstrating that the DBNs models presented in Chapter 5 are indeed acceptable as a foundation.

The PLE architecture that's proposed in this research is the vision of a future PLE. My belief is that PLE in the future has to be context-aware systems. The work which is presented in this research has been well researched, and laid out in a format which outlines each of my points in a clear manner.

### **6.2. Contributions**

This research presents two main contributions, which are presented below:

- An approach known as “Context-Aware Personal Learning Environment Architecture” is developed. The PLE architecture is automatically responsive to each individual user on demand, and it is able to predict user preferences or interests. The automated refinement of user requests for e-resources enables results to improve over time. Furthermore, the user is able to update the system by giving feedback to their PLE system in order to improve the result over time. The architecture consists of two layers: the top layer is the PLE service and the PLE interface for the user and the bottom layer shows other independent tools or

## Chapter 6: Conclusion and future work

service providers as selected by the user. This contribution was illustrated in Chapter 3, where comprehensive definitions of its components were presented.

- The implementation of the Prediction component in the proposed approach is presented using Bayesian Network techniques. Two DBN models are presented in the implementation. The first is used to predict a suitable provider for the user. The second model presented uses keywords from the Computer Science vocabulary to predict a suitable subject for the user. During the simulation of the DBN models, two time step counts were used in order to provide the previous and current states of the node. DBN models were able to obtain information from various sensors. With more simulation, the models become more confident. This contribution was presented in Chapter 5 and the simulation for the implementation was presented as part of Chapter 6.

### **6.3. Success criteria revisited**

The criteria for success were established in Chapter 1. This section focus on these criteria in order to determine the degree to which the research has been successful, as follows:

- To answer the research questions, a context-aware Personal Learning Environment architecture (context-aware PLE) has been built (Chapter 3). The architecture proposed in this research incorporates a context-aware engine. Thus there is intelligence built into the architecture and thus the PLE architecture is automatically responsive to the context information.
- To show how the proposed architecture is different from other PLE architectures, a comparative study with other PLE architectures has been presented demonstrating that the approach is better than that of other PLE architectures (Chapter 5). In summarising all the comparisons, it is apparent that the context-aware PLE approach is by far the strongest, as it contains 8 out of the 9 features, and is still under development.

## Chapter 6: Conclusion and future work

- To explain why a Bayesian Network was chosen, a study was carried out to inspect the currently available reasoning techniques. As a result, the BN technique was chosen from among the available ones (Chapter 4). The reasons why a BN has been chosen to implement the Prediction component in the proposed architecture are presented in Chapter 4.
- An implementation of the Prediction component using a Bayesian Network must be presented (Chapter 5). Two DBN models are presented in chapter 5. The first DBN model is used to predict a suitable provider for the user. The second model presented uses keywords from the Computer Science vocabulary to predict a suitable subject for the user. Both DBN models are found to be successfully valid.

### **6.4. Future work**

PLEs are still an active research area in the field of e-learning. Many issues remain to be considered. The following offer scope for extending this research:

- According to the research conducted in this thesis, the first track for future work is to extend the implementation of the proposed architecture. This research focuses on the Prediction component within the Generic subsystem. However, there are other components that should be implemented, such as Acquisition, Context History, Detector and Similarity components.
- Future work can be undertaken on the case study to include other PLE aspects, such as business and social aspects. A prime example being of how a business man would use the system to track down previous offers made between businesses, which then allows him to evaluate different avenues and then pursue which ever route suits him best. On the social front, it could be by implementing the preferences of the user, instead of using the proposed preferences which may not suit the user.

## Chapter 6: Conclusion and future work

- Integration of the proposed approach with different service providers should be investigated further, for example, how the proposal architecture connects with IEEE and ACM. The system must therefore have an Application Program Interface (API) to interact with the independent service provider.
- Currently, when a user conducts a research in Google, the result is often too many e-resources. To solve this problem, and one possible way in which this work could be used in the future, Google could adopt our system in presenting the search results. If Google does indeed use this system, then each user should receive only those results that are relevant to them.

## References

### **References**

- [1] M. Berry, "An Investigation of the Effectiveness of Moodle in Primary Education," *in Proceeding of IADIS Internet Conference*, pp. - 51-58, 2008.
- [2] A. Chavan and S. Pavri, "Open Source Learning Management in Moodle," *in: Linux Journal*, 1, No. 2, pp. 78-97, 2004.
- [3] E. Van Raaij and J. Schepers, "The acceptance and use of a virtual learning environment in China," *Computer Education.*, vol. 50, pp. 838-852, 2008.
- [4] A. Al-Ajlan, "Service oriented computing for dynamic virtual learning environments," Ph.D. De Montfort University, 2009.
- [5] C. Y. Cheng and J. Yen, "Virtual learning environment (VLE): A web-based collaborative learning system," in *System Sciences, 1998. Proceedings of the Thirty-First Hawaii International Conference on*, 1998, pp. 480-491.
- [6] A. Kumar, et al., "The virtual learning environment system," in *Frontiers in Education Conference, 1998. FIE'98. 28th Annual*, 1998, pp. 711-716.
- [7] B. Taraghi, et al., "Personal Learning Environment-a Conceptual Study," *International Conference on Interactive Computer Aided Learning*, vol. 5, pp. 25-30, 2010.
- [8] H. Zhao, et al., "The personal learning environment (PLE) based on web 2. 0," in *Web Society (SWS), 2010 IEEE 2nd Symposium on*, , pp. 22-25, 2010
- [9] K. Zubrinic and D. Kalpic, "The Web as Personal Learning Environment," *International Journal of Emerging Technologies in Learning*, vol. 3, 2008.
- [10] N. Sclater, "Web 2.0, Personal Learning Environments, and the Future of Learning Management Systems," in *Research Bulletin, Boulder*, vol. 13, CO: EDUCAUSE Center for Applied Research, 2008.



## References

- [11] S. Green, et al., "Formal specification of an adaptable personal learning environment using prolog," in *Proceedings of the 1<sup>st</sup> ACM SIGMM International Workshop on Media Studies and Implementations that Help Improving Access to Disabled Users*, pp. 29-38, 2009.
- [12] F. Wang, et al., "Construct personal learning environment based on Web 2. 0," *International Conference on Management and Service Science*, pp. 1-4, 2009.
- [13] D. Loertscher and C. Koechlin, "Personal Learning Environments in the Learning Commons," *Teacher Librarian*, vol. 39, pp. 23-26, 2011.
- [14] Educause Learning Initiative (ELI) (2009). The seven things you should know about Personal Learning Environments. Available. <http://net.educause.edu/ir/library/pdf/ELI7049.Pdf>, [15<sup>th</sup> Marc 2012].
- [15] G. Attwell, "Personal Learning Environments-the future of eLearning?" *ELearning Papers*, vol. 2, pp. 1-8, 2007.
- [16] S. Downes, "The Requirements of PLE,". Available: [http:// www.downes. ca/ post /57911](http://www.downes.ca/post/57911) ,[17<sup>th</sup> January 2012].
- [17] M. Amine, "on-going research on Knowledge and Learning: Requirements of a PLE Framework,". Available: [http:// mohamedaminechatti. blogspot. co.uk/2008/02/requirements-of-ple-framework.html](http://mohamedaminechatti.blogspot.co.uk/2008/02/requirements-of-ple-framework.html)[12<sup>th</sup> February 2013].
- [18] B. Olivier and O. Liber, "Lifelong Learning: The Need for Portable Personal Learning Environments and Supporting Interoperability Standards"; in *The JISC Centre for Educational Technology Interoperability Standards*, Bolton Institute, Bristol, 2001.
- [19] M. Harmelen, "Personal Learning Environments," In *Proc. Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies*, IEEE Computer Society, pp. 815-816, 2006.
- [20] O. Liber, "Colloquia—a conversation manager," *Campus-Wide Information Systems*, vol. 17, pp. 56-62, 2000.

## References

- [21] M. Harmelen, "The Manchester PLE Project," *JISC Emerge Benefits Realisation*, 2013, <https://www.TheManchesterPLEProject> [21/11/2013].
- [22] M. Harrison, et al., "Inclusive, adaptive design for students with learning disabilities," *International Conference on Advanced Learning Technologies*, 2008. *ICALT'08. Eighth IEEE*, pp. 1023-1027, 2008.
- [23] E. Pearson, et al., "From a personal learning environment to an adaptable personal learning environment," *10th International Conference on Advanced Learning Technologies (ICALT)*, 2010 *IEEE*, pp. 333-335, 2010.
- [24] S. Wilson, "Future VLE-The visual version," *Scott's Work blog*, 2005, <http://zope.cetis.ac.uk/members/scott/entries/2005012517020> [9th February, 2011].
- [25] S. Wilson, et al., "Personal Learning Environments: Challenging the dominant design of educational systems," *Journal of e-Learning and Knowledge Society-English Version*, vol. 3, 2009.
- [26] M. Chatti, "Toward a personal learning environment framework," *International Journal of Virtual and Personal Learning Environments (IJVPLE)*, vol. 1, pp. 66-85, 2010.
- [27] S. Murugesan, "Understanding Web 2.0," *IT Professional*, vol. 9, pp. 34-41, 2007.
- [28] S. Hoare, "Students tell universities: Get out of MySpace!". *Education Guardian*, vol. 5, 2007.
- [29] S. Loke, *Context-Aware Pervasive Systems: Architectures for a New Breed of Applications*. CRC Press, 2006.
- [30] M. Weiser. Available:<http://pubweb.parc.xerox.com/weiser/weiser.html> [ 6/ 17/ 2011].
- [31] B. Schilit and M. Theimer, "Disseminating active map information to mobile hosts," *Network, IEEE*, vol. 8, pp. 22-32, 1994.

## References

- [32] *Dictionary.com – Free Online English Dictionary*. Available: [http:// dictionary.reference.com/](http://dictionary.reference.com/) [4/16/2011].
- [33] B. Schilit, et al., "Context-aware computing applications," *First Workshop on Mobile Computing Systems and Applications*, " WMCSA, pp. 85-90, 1994.
- [34] M. Baldauf, et al., "A survey on context-aware systems," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, pp. 263-277, 2007.
- [35] A. K. Dey, "Understanding and using context," *Personal and Ubiquitous Computing*, vol. 5, pp. 4-7, 2001.
- [36] A. Zimmermann, et al." An operational definition of context" in *Preceding of Modelling and using Context* Anonymous, Springer, pp. 558-571, 2007, p.p. 558-571, 2007.
- [37] J. Pascoe, "Adding generic contextual capabilities to wearable computers," in *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*, 1998, pp. 92-99.
- [38] G. D. Abowd, et al. "Towards a better understanding of context and context-awareness," in *Handheld and Ubiquitous Computing*, 1999, pp. 304-307.
- [39] K. K. Khedo, "Context-aware systems for mobile and ubiquitous networks," *International Conference on Mobile Communications and Learning Technologies, 2006. ICN/ICONS/MCL 2006*, pp. 123-123, 2006.
- [40] A. Hristova, "Conceptualization and Design of a Context-aware Platform for user centric Applications," master's thesis Norwegian University of Science and Technology, Technical University of Madrid, 2008.
- [41] D. Zhang, et al., "Context-aware infrastructure for personalized healthcare," *Stud. Health Technol. Inform.*, vol. 117, pp. 154-163, 2005.
- [42] J. Indulska and P. Sutton, "Location management in pervasive systems," in *Proceedings of the Australasian Information Security Workshop Conference on ACSW Frontiers 2003-Volume 21*, 2003, pp. 143-151.

## References

- [43] A. Schmidt and K. Van Laerhoven, "How to build smart appliances?" *Personal Communications, IEEE*, vol. 8, pp. 66-71, 2001.
- [44] H. Chen, et al., "An intelligent broker for context-aware systems," in *Adjunct Proceedings of Ubicomp*, 2003, pp. 183-184.
- [45] M. Hong and D. Cho, "Ontology context model for context-aware learning service in ubiquitous learning environments," *International Journal of Computers*, vol. 2, pp. 172-178, 2008.
- [46] Y. Oh, et al., "Designing, developing, and evaluating context-aware systems," *International Conference on in Multimedia and Ubiquitous Engineering*, 2007. *MUE'07.*, 2007, pp. 1158-1163.
- [47] M. Al-Sammarraie, "Policy-based approach for context-aware systems," Ph.D. De Montfort University, 2011.
- [48] F. Siewe, H. Zedan and A. Cau, "The calculus of context-aware ambient," *Journal of Computer and System Sciences*, vol. 77, pp. 597-620, 2011.
- [49] C. Bettini, et al., "A survey of context modelling and reasoning techniques," *Pervasive and Mobile Computing*, vol. 6, pp. 161-180, 2010.
- [50] T. Strang and C. Linnhoff-Popien, "A context modelling survey," in *Workshop Proceedings*, 2004.
- [51] H. Ailisto, et al., "Structuring context aware applications: Five-layer model and example case," in *Proceedings of the Workshop on Concepts and Models for Ubiquitous Computing*, pp. 1-5, 2002.
- [52] O. Kwon, S. Choi and G. Park, "NAMA: a context-aware multi-agent based web service approach to proactive need identification for personalized reminder systems," *Expert Syst. Appl.*, vol. 29, pp. 17-32, 2005.
- [53] K. Cheverst, N. Davies, K. Mitchell and P. Smith, "Providing tailored (context-aware) information to city visitors," in *Adaptive Hypermedia and Adaptive Web-Based Systems*, 2000, pp. 73-85.

## References

- [54] R. Want, A. Hopper, V. Falcao and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems (TOIS)*, vol. 10, pp. 91-102, 1992.
- [55] M. Al-Doori, "Directional routing techniques in VANET," Ph.D. De Montfort University, 2011.
- [56] S. Stojanov, et al., "DeLC-distributed eLearning center," in *1st Balkan Conference on Informatics (BCI'2003)*, Thessaloniki, Greece, 2003.
- [57] H. Dali and G. Yongmei, "Analysis on the coupling between E-learning and lifelong learning," in *Computer Science and Software Engineering, 2008 International Conference on*, 2008, pp. 246-249.
- [58] A. Volodina, "Evaluation in lifelong learning," *14th International Conference on Interactive Collaborative Learning (ICL)*, 2011, pp. 1-4, 2011.
- [59] E. B. Wilson, *an Introduction to Scientific Research*. Courier Dover Publications, 1990.
- [60] M. Negnevitsky, *Artificial Intelligence: A Guide to Intelligent Systems*. Pearson Education, 2005.
- [61] C. Bettini, et al. "A survey of context modelling and reasoning techniques," *Pervasive and Mobile Computing*, vol. 6, pp. 161-180, 2010.
- [62] A. A. Hopgood, *Intelligent Systems for Engineers and Scientists*. CRC press, 2011.
- [63] H. Wu, et al. "Sensor fusion using dempstershafer theory, for context-aware HCI," *Conference in Instrumentation and Measurement Technology, 2002. IMTC/2002. Proceedings of the 19th IEEE*, pp. 7-12, 2002.
- [64] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford university press, 1995.
- [65] Neural networks. Available :[http:// www. eal.ei.tum. de/fileadmin /tueieal/ www /courses /PE/lecture/Guests/Munich 2012. Presentation.pdf](http://www.eal.ei.tum.de/fileadmin/tueieal/www/courses/PE/lecture/Guests/Munich%2012/Presentation.pdf). [2/3/2013].

## References

- [66] F. Tapia, et al., "Bayesian Model for Optimization Adaptive E-Learning Process." *International Journal of Emerging Technologies in Learning*, vol. 3, 2008.
- [67] Limitations of Fuzzy Logic. Available: <http://www.ehow.com/info12216963limitations-fuzzy-logic.html> [3/13/2013].
- [68] S. J. Al-Sultan, "Context Aware Drivers' Behaviour Detection System for VANET," Ph.D. De Montfort University, 2013.
- [69] J. Hulst, "Modeling physiological processes with dynamic Bayesian networks," *Faculty of Electrical Engineering, Mathematics, and Computer Science, University of Pittsburgh*, 2006.
- [70] Q. Ji, et al., "A probabilistic framework for modelling and real-time monitoring human fatigue," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 36, pp. 862-875, 2006.
- [71] S. Ramachandran and R. J. Mooney, *Theory Refinement of Bayesian Networks with Hidden Variables*, 1998.
- [72] V. Mihajlovic and M. Petkovic, "Dynamic bayesian networks: A state of the art," 2001.
- [73] K. B. Korb and A. E. Nicholson, *Bayesian Artificial Intelligence*. cRc Press, 2003.
- [74] N. Friedman, "Learning Bayesian networks in the presence of missing values and hidden variables," in *Proc. of the Conf. on Uncertainty in AI*, pp. 125-133, 1997.
- [75] X. Li and Q. Ji, "Active affective state detection and user assistance with dynamic Bayesian networks," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 35, pp. 93-105, 2005.
- [76] D. Heckerman, *A Tutorial on Learning with Bayesian Networks*. Springer, 1998.
- [77] D. Heckerman, C. Meek and G. Cooper, "A Bayesian approach to causal discovery," *Computation, Causation, and Discovery*, vol. 19, pp. 141-166, 1999.

## References

- [78] M. Xenos, "Prediction and assessment of student behaviour in open and distance education in computers using Bayesian networks," *Comput. Educ.*, vol. 43, pp. 345-359, 2004.
- [79] O. Pourret et al., "Bayesian Networks: A Practical Guide to Applications," John Wiley & Sons, 2008.
- [80] T. Koski and J. Noble, "Bayesian Networks: An Introduction," John Wiley & Sons, 2011.
- [81] H. Guo and W. Hsu, "A survey of algorithms for real-time bayesian network inference," in *AAAI/KDD/UAI02 Joint Workshop on Real-Time Decision Support and Diagnosis Systems*, 2002.
- [82] D. Chickering et al., "Learning Bayesian Networks is NP-Hard," *Technical Report MSR-TR-94-17, Microsoft Research, Microsoft Corporation*, 1994.
- [83] B. A. Truong, Y. Lee and S. Lee, "Modeling and reasoning about uncertainty in context-aware systems," in *E-Business Engineering, 2005. ICEBE 2005. IEEE International Conference on*, 2005, pp. 102-109.
- [84] X. Li and Q. Ji, "Active affective state detection and user assistance with dynamic Bayesian networks," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 35, pp. 93-105, 2005.
- [85] T. Stephenson, "An introduction to Bayesian network theory and usage," *Research report, idiap-rr 00-03, Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP)*, 2000.
- [86] J. Pearl, "Probabilistic Reasoning in Intelligent Systems, " *Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- [87] P. Youguo, et al., "The frame of cognitive pattern recognition," in *Control Conference. Chinese*, 2007, pp. 694-696.
- [88] G. F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Mach. Learning*, vol. 9, pp. 309-347, 1992.

## References

- [89] S. Nedeveschi, J. S. Sandhu, J. Pal, R. Fonseca and K. Toyama, "Bayesian networks: An exploratory tool for understanding ICT adoption," in *Information and Communication Technologies and Development, 2006. ICTD'06. International Conference on*, 2006, pp. 277-284.
- [90] S. Nadkarni and P. P. Shenoy, "A causal mapping approach to constructing Bayesian networks," *Decis. Support Syst.*, vol. 38, pp. 259-281, 2004.
- [91] Y. Tong and Q. Ji, "Learning bayesian networks with qualitative constraints," *IEEE Conference on Computer Vision and Pattern Recognition. CVPR*, pp. 1-8, 2008.
- [92] P. Garcia, S. Schiaffino and A. Amandi, "An enhanced Bayesian model to detect students' learning styles in Web-based courses," *J. Comput. Assisted Learn.*, vol. 24, pp. 305-315, 2008.
- [93] G. Yang, Y. Lin and P. Bhattacharya, "A driver fatigue recognition model based on information fusion and dynamic Bayesian network," *Inf. Sci.*, vol. 180, pp. 1942-1954, 2010.
- [94] D. Rao and S. Mangalwede, "Application of Bayesian Networks for Learner Assessment in E-Learning Systems". *International Journal of Computer Applications*, 4(4), pp. 23–28, July 2010.
- [95] I. Ben-Gal, "Bayesian Networks," in Ruggeri, F., Kenett, R., Faltin, F. (editors): *Encyclopedia of Statistics in Quality and Reliability*, John Wiley & Sons, 2007.
- [96] H. Si, Y. Kawahara, H. Morikawa and T. Aoyama, "A stochastic approach for creating context-aware services based on context histories in smart home," *Cognitive Science Research Paper-University of Sussex Csrp*, vol. 577, pp. 37, 2005.
- [97] Y. Zhang, Q. Ji and C. G. Looney, "Active information fusion for decision making under uncertainty," in, 2002. *Proceedings of the Fifth International Conference on Information Fusion*, pp. 643-650, 2002.



## References

- [98] K. Murphy and S. Mian, "Modeling gene expression data using dynamic Bayesian networks", [Online]. University of California, Berkeley. <http://www.cs.berkeley.edu/murphyk/publ.html> [21/3/2013].
- [99] V. Pavlovic, "Dynamic Bayesian Networks for Information Fusion with Application to Human-Computer Interfaces," *PhD dissertation, Dept. of Electrical and Computer Eng.*, Univ. of Illinois at Urbana-Champaign, 1999.
- [100] Bayesia - BayesiaLab: The technology of bayesian networks at your service. Available: <http://www.bayesia.com/en/products/bayesialab.php> [5/13/2013].
- [101] *MATLAB* - The Language of Technical Computing- *MathWorks United Kingdom*. Available: <http://www.mathworks.co.uk/products/matlab/> [9/18/2013].
- [102] DSL . Available: <http://genie.sis.pitt.edu/wiki/MainPage> [10/13/2013].
- [103] Ching-Bang Yao, "Applied bayesian network to construct the adaptive personalized e-learning platform," in *Computer and Automation Engineering (ICCAE), 2010 the 2nd International Conference on*, pp. 857-861, 2010.
- [104] K. KrishnaKumar," Intelligent Systems for Aerospace Engineering - An Overview". *Proceedings of the RTO AVT Course on Intelligent Systems for Aeronautics*. Belgium, May 2003.
- [105] M. T. Alharbi, A. Platt and A. H. Al-Bayatti, "Context-aware personal learning environment," *International Conference for Internet Technology and Secured Transactions*, London, pp. 692-697, 2012.
- [106] M. T. Alharbi, A. Platt and A. H. Al-Bayatti, "Personal Learning Environment," *International Journal for e-Learning Security*, vol.4, Issue 1/2, 2013.
- [107] W. Alkhaldi, et al.," Development of Context-Aware Advertisement Systems". *International Conference on Computer Applications and Network Security*, Maldives, May 2011.

## References

[108] M. Weiser, "The computer for the 21st century," *Sci. Amer*, vol.265, pp. 66–75, Sept. 1991.

## Appendices

### Appendix A: Survey

This survey consists of two parts, comprises 14 questions in all. The target respondents are research students in the field of technology in the UK [180 respondents participated in the data collection process]. The first part has 9 questions, which consider the factors that affect students when selecting providers. The second part consists of 5 questions, and focuses on computer science keywords as well as other considerations that affect computer science. All of the questions for both parts are presented below:

First part: Provider:

- 1- Which one of the following providers does your supervisor advise you to use?
  - Ethos
  - Library
  - IEEE
- 2- When you look for e-resources (information), do you need details?
  - Yes
  - No
- 3- Which type of material do you normally use?
  - Thesis
  - Book
  - Conference paper
- 4- Do you use free resources?
  - Yes
  - No
- 5- Which one of the following providers are you a member of or have an account with?
  - Ethos
  - Library
- 6- Which one of the following providers do you use?
  - Ethos
  - Library
  - IEEE

## Appendices

8- Which year of study are you in?

- First
- Second
- Third

9- When you look for e-resources (information), do you need general information?

- Yes
- No

### **Part 2:** Computer science keyword

1- The following topics are related to computer science; please select the topic relevant to you.

- Software engineering
- Computer security
- Artificial intelligence

2- If you are interested in artificial intelligence, which of these sub-titles are you interested in?

- Computational modelling
- Mobile robots

3- If you are interest in computer security, which of these sub-titles are you interested in?

- Forensic computing
- Security management

4- If you are interested in software engineering, which of these sub-titles are you interested in?

- Formal methods
- Requirements engineering

5- Please select the relevant field; you can choose more than one.

- Dynamic systems
- Fuzzy logic
- Digital evidence

## Appendices

- Computer crime
- UML
- Z notation

## Appendices

### Appendix B: Transfer data file to BN software

There are certain steps to follow in transferring data files to the Bayesian Network software. These steps are as follows:

Step 1: Generate a data file in GeNIe as shown in Figure B.1:

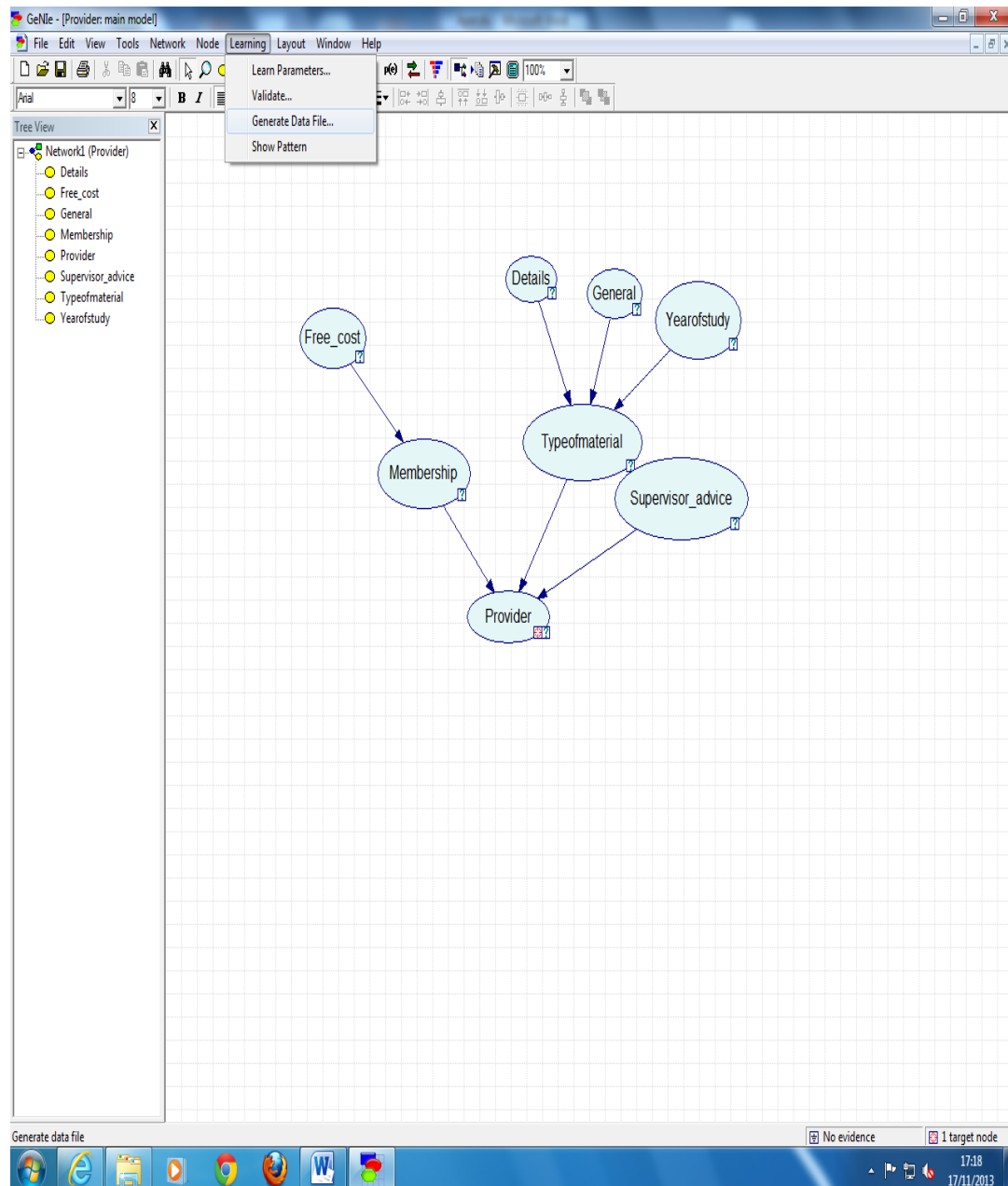


Figure B.1: Generate a data file in GeNIe, step 1

## Appendices

Step 2: Enter the number of the participants as shown in Figure B.2; the number is 180.

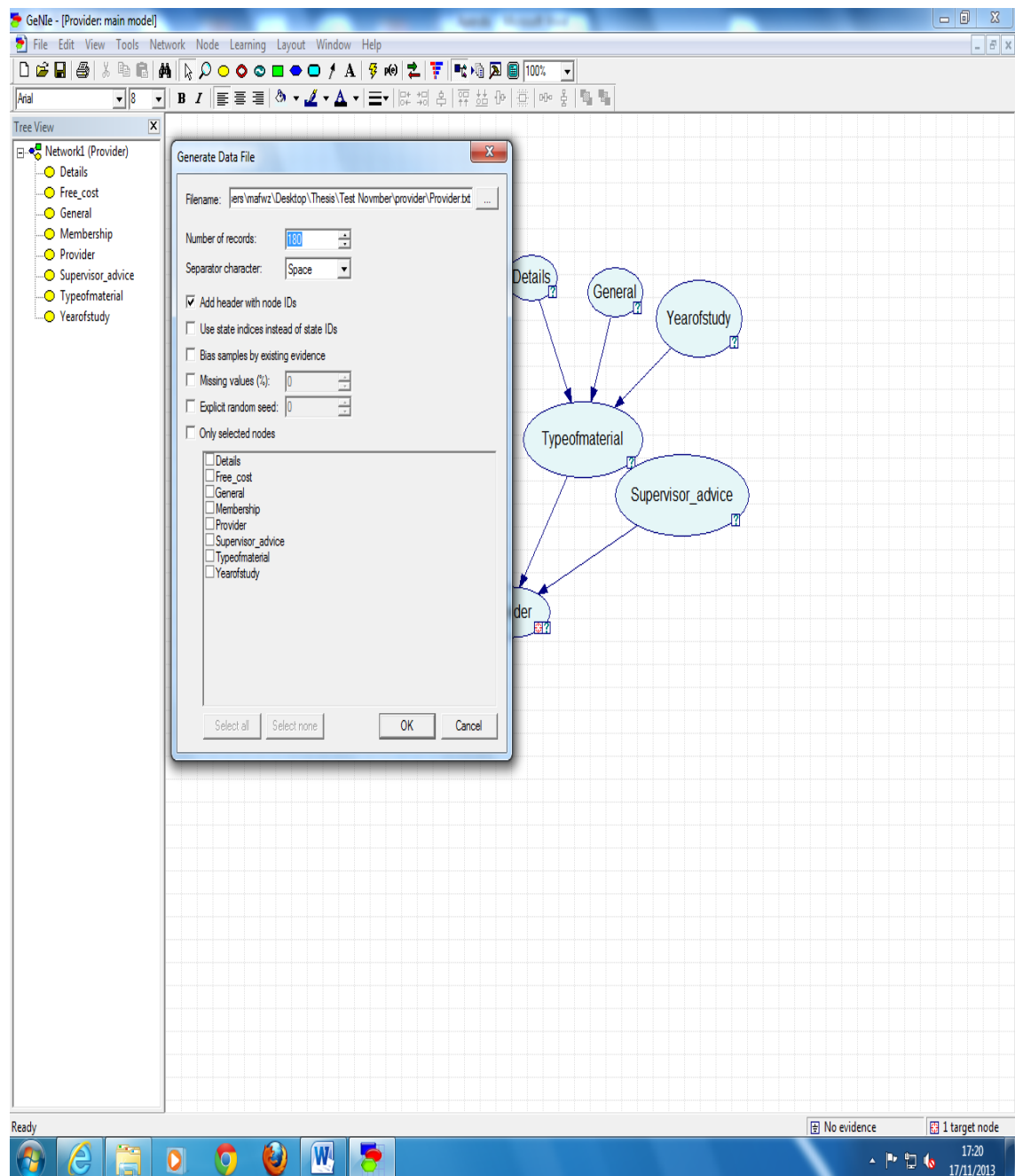


Figure B.2: Generate a data file in GeNIe, step 2

## Appendices

Step 3: GeNIe will be able to create the data file as shown in Figure B.3; the data file is created and there are two ways to open it (through GeNIe or Notepad).

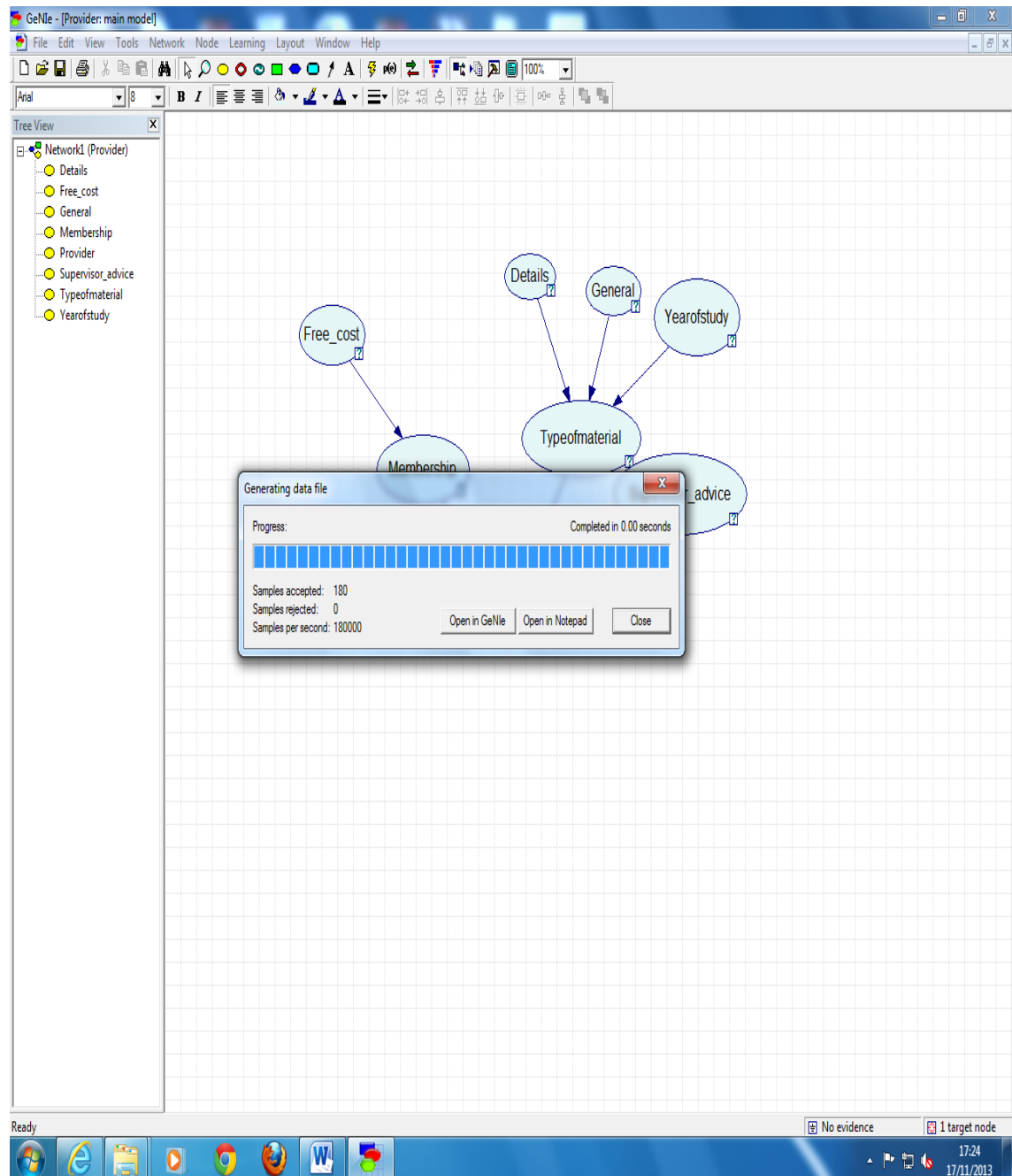
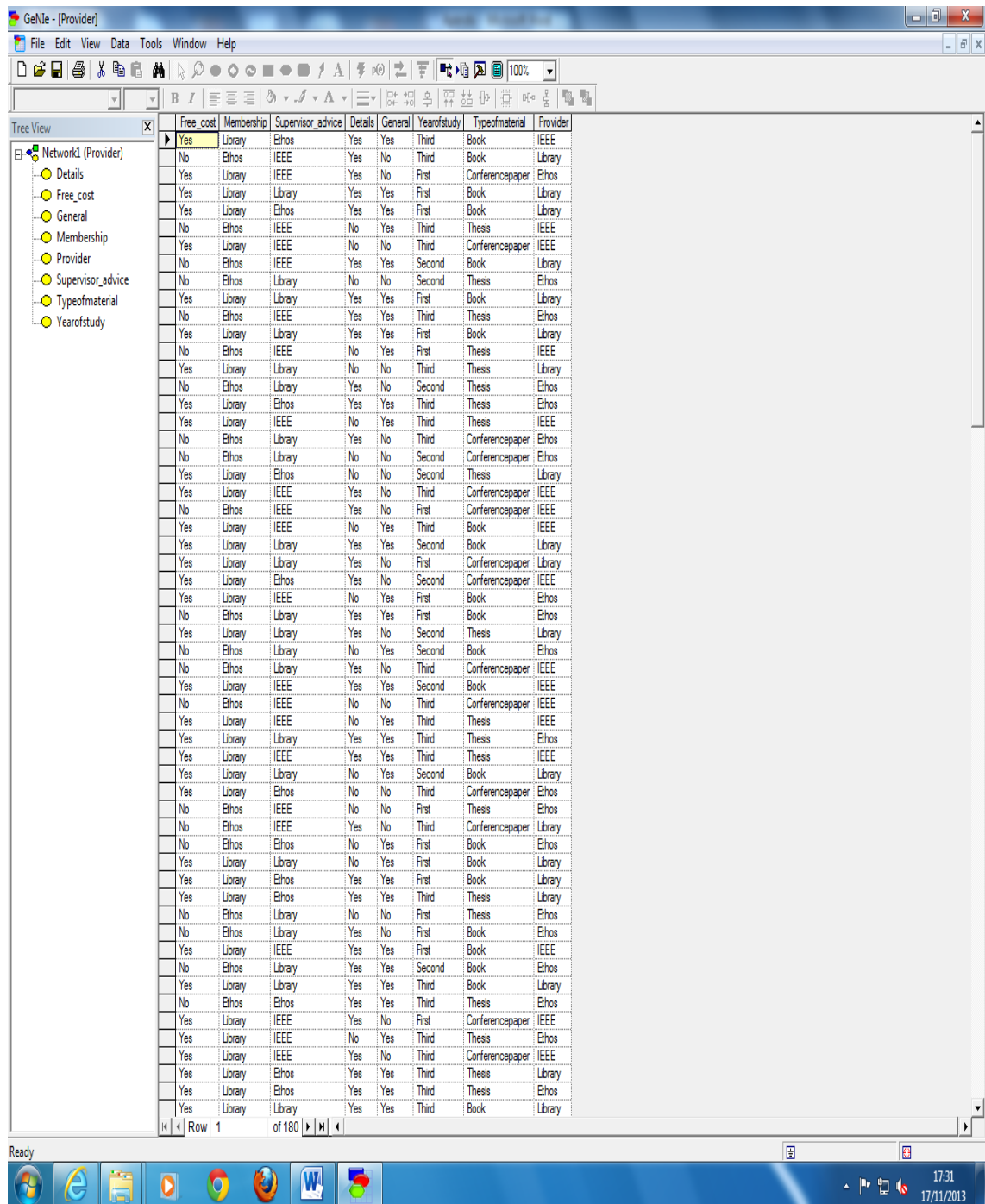


Figure B.3: Generate a data file in GeNIe, step 3



## Appendices

Step 4: Figure B.4 below shows that the data file is open in GeNIe; however, these data are created randomly by the software.



Free_cost	Membership	Supervisor_advice	Details	General	Yearofstudy	Typeofmaterial	Provider
Yes	Library	Ethos	Yes	Yes	Third	Book	IEEE
No	Ethos	IEEE	Yes	No	Third	Book	Library
Yes	Library	IEEE	Yes	No	First	Conferencepaper	Ethos
Yes	Library	Library	Yes	Yes	First	Book	Library
Yes	Library	Ethos	Yes	Yes	First	Book	Library
No	Ethos	IEEE	No	Yes	Third	Thesis	IEEE
Yes	Library	IEEE	No	No	Third	Conferencepaper	IEEE
No	Ethos	IEEE	Yes	Yes	Second	Book	Library
No	Ethos	Library	No	No	Second	Thesis	Ethos
Yes	Library	Library	Yes	Yes	First	Book	Library
No	Ethos	IEEE	Yes	Yes	Third	Thesis	Ethos
Yes	Library	Library	Yes	Yes	First	Book	Library
No	Ethos	IEEE	No	Yes	First	Thesis	IEEE
Yes	Library	Library	No	No	Third	Thesis	Library
No	Ethos	Library	Yes	No	Second	Thesis	Ethos
Yes	Library	Ethos	Yes	Yes	Third	Thesis	Ethos
Yes	Library	IEEE	No	Yes	Third	Thesis	IEEE
No	Ethos	Library	Yes	No	Third	Conferencepaper	Ethos
No	Ethos	Library	No	No	Second	Conferencepaper	Ethos
Yes	Library	Ethos	No	No	Second	Thesis	Library
Yes	Library	IEEE	Yes	No	Third	Conferencepaper	IEEE
No	Ethos	IEEE	Yes	No	First	Conferencepaper	IEEE
Yes	Library	IEEE	No	Yes	Third	Book	IEEE
Yes	Library	Library	Yes	Yes	Second	Book	Library
Yes	Library	Library	Yes	No	First	Conferencepaper	Library
Yes	Library	Ethos	Yes	No	Second	Conferencepaper	IEEE
Yes	Library	IEEE	No	Yes	First	Book	Ethos
No	Ethos	Library	Yes	Yes	First	Book	Ethos
Yes	Library	Library	Yes	No	Second	Thesis	Library
No	Ethos	Library	No	Yes	Second	Book	Ethos
No	Ethos	Library	Yes	No	Third	Conferencepaper	IEEE
Yes	Library	IEEE	Yes	Yes	Second	Book	IEEE
No	Ethos	IEEE	No	No	Third	Conferencepaper	IEEE
Yes	Library	IEEE	No	Yes	Third	Thesis	IEEE
Yes	Library	Library	Yes	Yes	Third	Thesis	Ethos
Yes	Library	IEEE	Yes	Yes	Third	Thesis	IEEE
Yes	Library	Library	No	Yes	Second	Book	Library
Yes	Library	Ethos	No	No	Third	Conferencepaper	Ethos
No	Ethos	IEEE	No	No	First	Thesis	Ethos
No	Ethos	IEEE	Yes	No	Third	Conferencepaper	Library
No	Ethos	Ethos	No	Yes	First	Book	Ethos
Yes	Library	Library	No	Yes	First	Book	Library
Yes	Library	Ethos	Yes	Yes	First	Book	Library
Yes	Library	Ethos	Yes	Yes	Third	Thesis	Library
No	Ethos	Library	No	No	First	Thesis	Ethos
No	Ethos	Library	Yes	No	First	Book	Ethos
Yes	Library	IEEE	Yes	Yes	First	Book	IEEE
No	Ethos	Library	Yes	Yes	Second	Book	Ethos
Yes	Library	Library	Yes	Yes	Third	Book	Library
No	Ethos	Ethos	Yes	Yes	Third	Thesis	Ethos
Yes	Library	IEEE	Yes	No	First	Conferencepaper	IEEE
Yes	Library	IEEE	No	Yes	Third	Thesis	Ethos
Yes	Library	IEEE	Yes	No	Third	Conferencepaper	IEEE
Yes	Library	Ethos	Yes	Yes	Third	Thesis	Library
Yes	Library	Ethos	Yes	Yes	Third	Thesis	Ethos
Yes	Library	Library	Yes	Yes	Third	Book	Library

Figure B.4: Generate a data file in GeNIe, step 4

## Appendices

## Appendix C: Data file (results from the survey)

- First part: Data file for the first model

GeNIe allows the developer to change the data file according to the answers. I have inserted all the responses from the survey into the data file, as shown in Figures C.1 - C4:

[illegible]

Figure C. 1: Data file for the first model, page 1.

[illegible]

145

[illegible]

146

[illegible]

147



## Appendices

- Second part: Data file for the second model.

Figures C5 - C7 show the original data file for the second model.

GeNIe - [Keyword]

File Edit View Data Tools Window Help

100%

Tree View

- Network1 (Keyword)
  - Artificial\_Intelligence
  - Computer\_Crime
  - Computer\_Science\_Key
  - Computer\_Security
  - Digital\_Evidence
  - Dynamic\_system
  - Fuzzy\_logic
  - Software\_Engineering
  - UML
  - Z\_notation

Digital_Evidence	Computer_Crime	Computer_Security	UML_Z_notation	Software_Engineering	Fuzzy_logic	Dynamic_system	Artificial_Intelligence	Computer_Science_Keyword
no	no	Forensic_Computing	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
no	no	Security_Management	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
no	no	Forensic_Computing	yes	Formal_Methods	no	no	Computational_Modeling	Software_Engineering
no	no	Forensic_Computing	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
no	no	Forensic_Computing	yes	Formal_Methods	no	no	Computational_Modeling	Software_Engineering
no	no	Security_Management	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
no	no	Security_Management	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
no	no	Forensic_Computing	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
no	no	Forensic_Computing	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
no	no	Security_Management	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
no	no	Security_Management	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
no	no	Security_Management	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
no	no	Forensic_Computing	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
no	no	Forensic_Computing	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
no	no	Security_Management	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
no	no	Security_Management	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
no	no	Forensic_Computing	yes	Formal_Methods	no	no	Computational_Modeling	Software_Engineering
no	no	Forensic_Computing	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
no	no	Forensic_Computing	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
no	no	Security_Management	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
no	no	Forensic_Computing	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
no	no	Security_Management	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
no	no	Security_Management	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
no	no	Security_Management	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
no	no	Forensic_Computing	no	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
no	no	Forensic_Computing	no	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
no	no	Security_Management	no	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
no	no	Security_Management	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
no	no	Security_Management	no	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
no	no	Forensic_Computing	no	Formal_Methods	no	no	Computational_Modeling	Software_Engineering
no	no	Forensic_Computing	no	Formal_Methods	no	no	Computational_Modeling	Software_Engineering
no	no	Forensic_Computing	no	Formal_Methods	no	no	Computational_Modeling	Software_Engineering
no	no	Forensic_Computing	no	Formal_Methods	no	no	Computational_Modeling	Software_Engineering
no	no	Forensic_Computing	no	Formal_Methods	no	no	Computational_Modeling	Software_Engineering
no	no	Forensic_Computing	no	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
no	no	Security_Management	no	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
no	yes	Forensic_Computing	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
no	yes	Forensic_Computing	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
no	yes	Forensic_Computing	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
no	yes	Forensic_Computing	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
no	yes	Forensic_Computing	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
no	yes	Forensic_Computing	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
no	yes	Forensic_Computing	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
no	yes	Forensic_Computing	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
no	yes	Forensic_Computing	no	Formal_Methods	no	no	Computational_Modeling	Software_Engineering
no	yes	Forensic_Computing	no	Formal_Methods	no	no	Computational_Modeling	Software_Engineering
no	yes	Forensic_Computing	no	Formal_Methods	no	no	Computational_Modeling	Software_Engineering
no	yes	Forensic_Computing	no	Formal_Methods	no	no	Computational_Modeling	Software_Engineering

Row 55 of 180

Ready Windows Media Player

17:51 17/11/2013

Figure C.5: Data file for the second model, page 1

## Appendices

GeNIe - [Keyword]

File Edit View Data Tools Window Help

100%

Tree View

- Network1 (Keyword)
  - Artificial\_Intelligence
  - Computer\_Crime
  - Computer\_Science\_Key
  - Computer\_Security
  - Digital\_Evidence
  - Dynamic\_system
  - Fuzzy\_logic
  - Software\_Engineering
  - UML
  - Z\_notation

	Digital_Evidence	Computer_Crime	Computer_Security	UML	Z_notation	Software_Engineering	Fuzzy_logic	Dynamic_system	Artificial_Intelligence	Computer_Science_Keyword
	no	yes	Forensic_Computing	no	yes	Formal_Methods	no	no	Computational_Modeling	Software_Engineering
	no	yes	Forensic_Computing	no	yes	Formal_Methods	no	no	Computational_Modeling	Software_Engineering
	no	yes	Forensic_Computing	no	yes	Formal_Methods	no	no	Computational_Modeling	Software_Engineering
	no	yes	Forensic_Computing	no	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Forensic_Computing	no	no	Formal_Methods	yes	yes	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	yes	yes	Mobile_Robotics	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	yes	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	yes	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Formal_Methods	no	no	Computational_Modeling	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
	no	no	Security_Management	yes	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	yes	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
	no	no	Security_Management	yes	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	yes	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
	no	no	Security_Management	yes	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Formal_Methods	no	no	Computational_Modeling	Software_Engineering
	no	no	Security_Management	yes	yes	Formal_Methods	no	no	Computational_Modeling	Software_Engineering
	no	no	Security_Management	yes	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	yes	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Formal_Methods	no	no	Computational_Modeling	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	yes	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
	no	no	Security_Management	yes	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	yes	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	yes	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Formal_Methods	no	no	Computational_Modeling	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	yes	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
	no	no	Security_Management	yes	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	yes	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	yes	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
	no	no	Security_Management	yes	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
	no	no	Security_Management	yes	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	yes	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	yes	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
	no	no	Forensic_Computing	yes	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Forensic_Computing	no	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
	no	no	Forensic_Computing	no	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
	no	no	Security_Management	no	yes	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
	no	no	Security_Management	no	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	yes	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	yes	Requirement_Engineering	no	no	Mobile_Robotics	Software_Engineering
	no	no	Forensic_Computing	no	no	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
	no	no	Forensic_Computing	no	no	Requirement_Engineering	no	no	Computational_Modeling	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods	no	no	Mobile_Robotics	Software_Engineering
	no	no	Security_Management	no	no	Formal_Methods				

Figure C.6: Data file for the second model, page 2

## Appendices

[illegible]

Figure C.7: Data file for the second model, page 3



## Appendices

### Appendix D: Creating DBN

Step 1: To create the DBN, the first step is that an SBN must be created, as shown in Figure D.1.

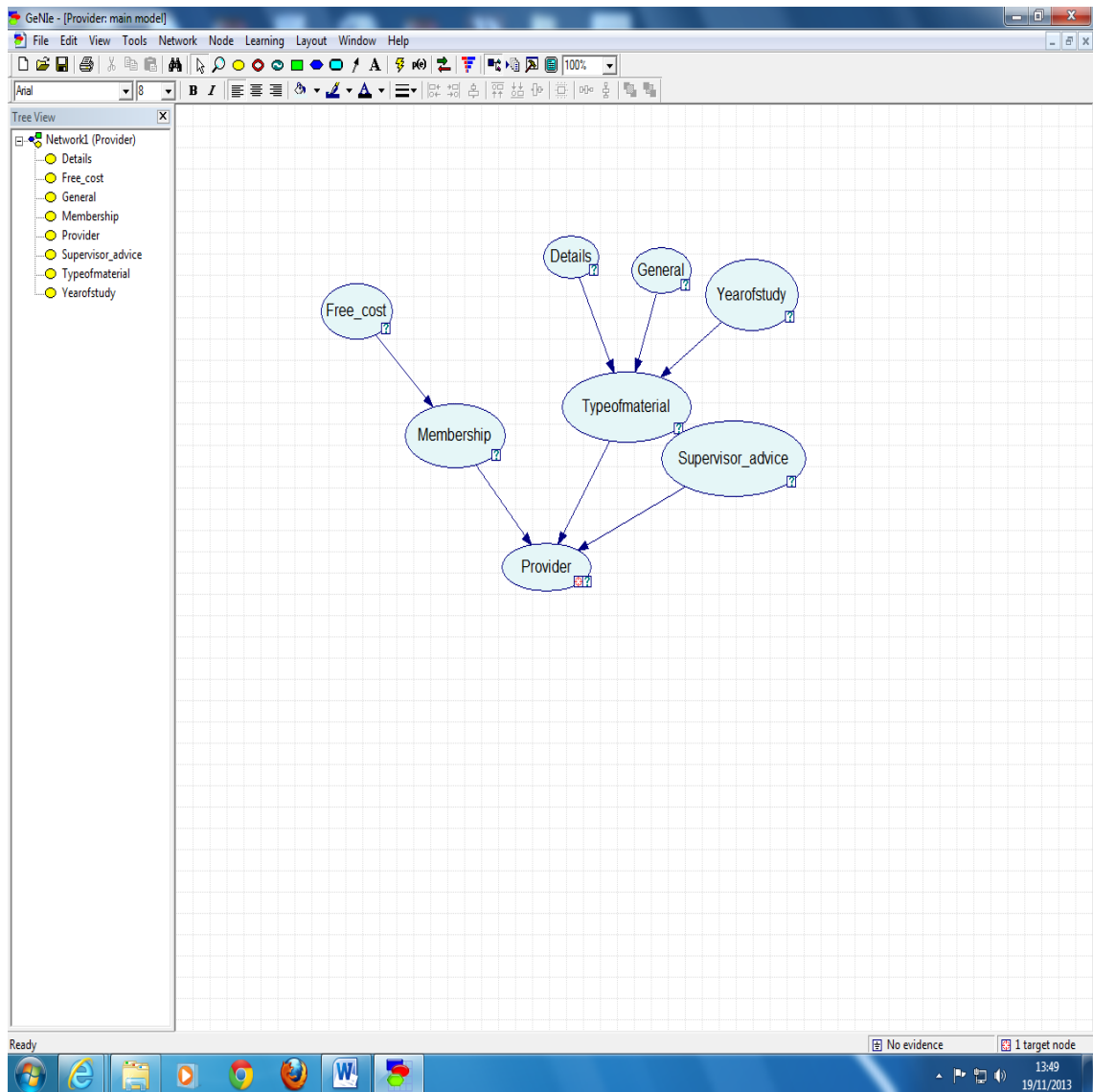


Figure D.1: Creating DBN, step 1

## Appendices

Step 2: A temporal plate must be activated; this is because it is a temporal network (as shown in Figure D.2).

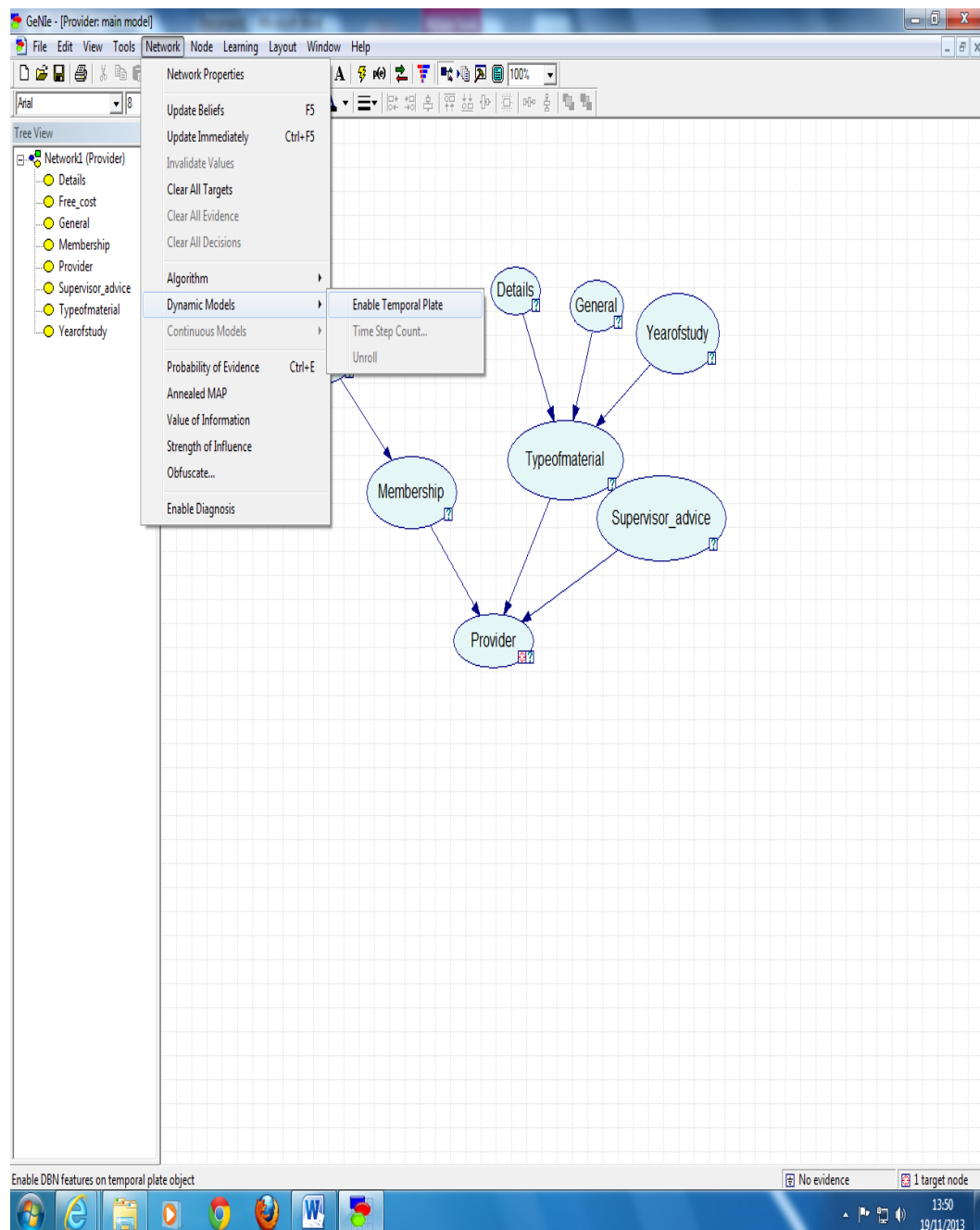


Figure D.2: Creating DBN, step 2

## Appendices

Step 3: The temporal plate will be presented as shown in Figure D.3 .

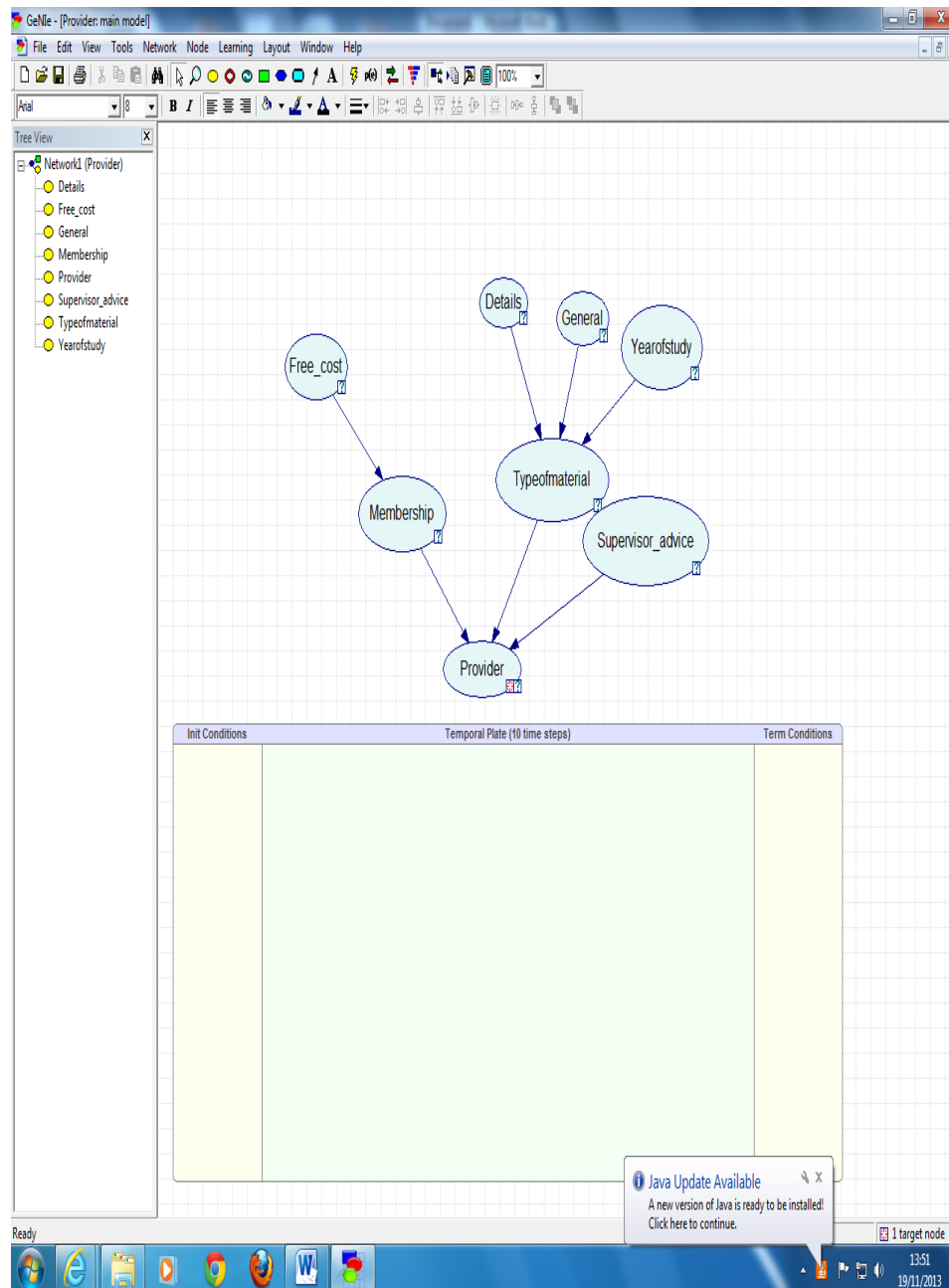


Figure D.3: Creating DBN, step 3

## Appendices

Step 4: Select the time step count; Figures D.4 and D.5.

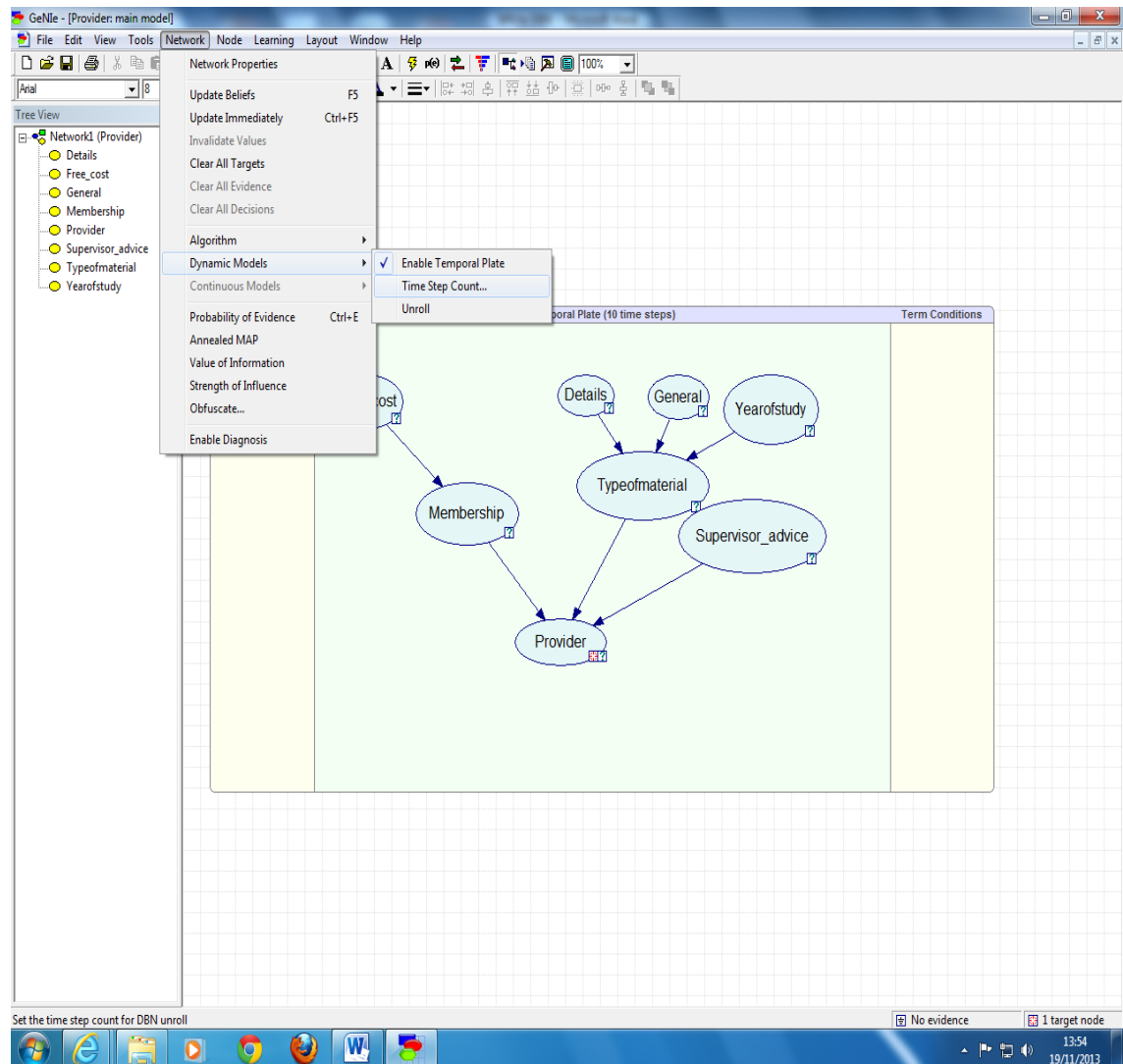


Figure D.4: Creating DBN, step 4 A

## Appendices

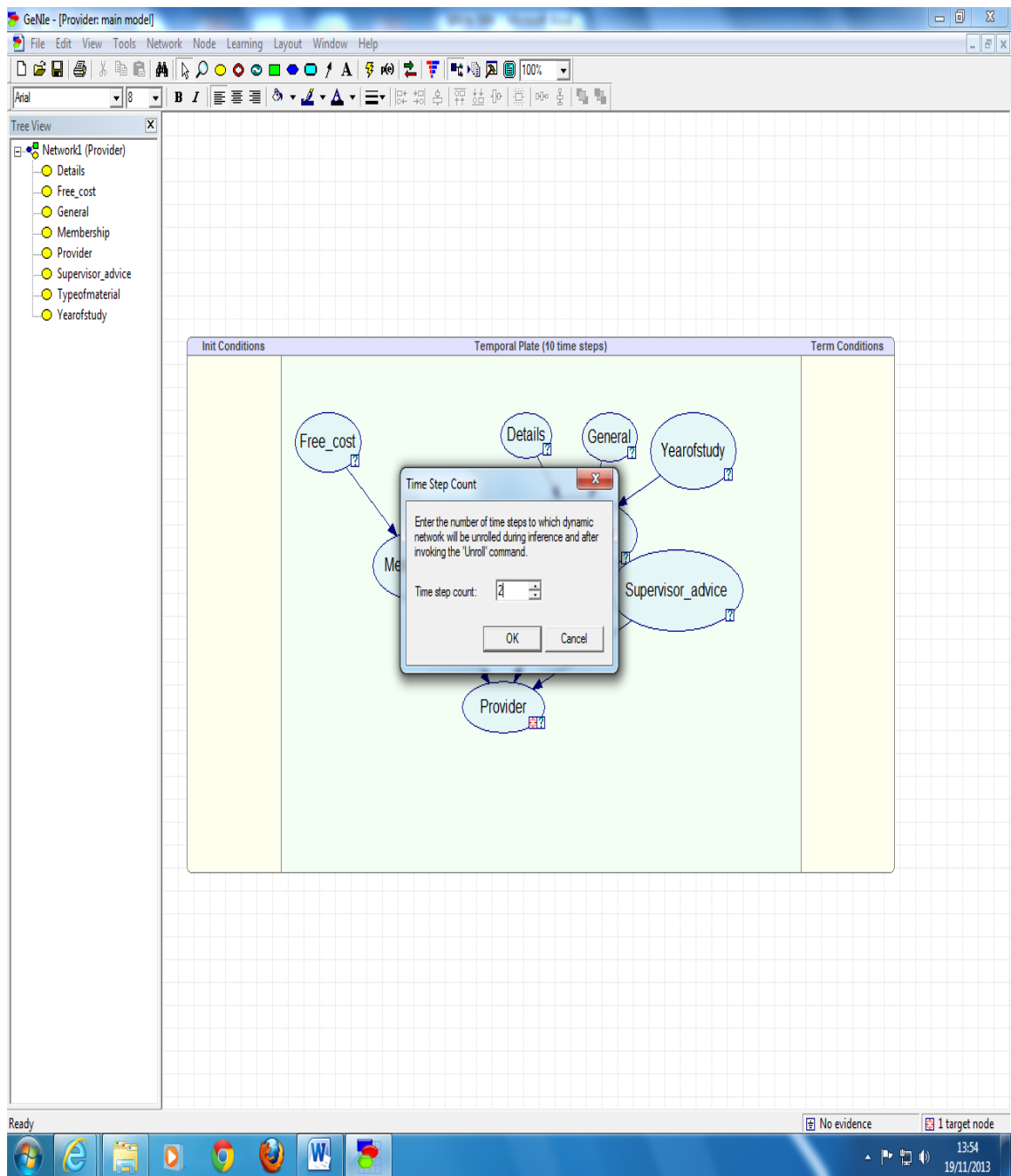


Figure D.5: Creating DBN, step 4 B

## Appendices

Step 5: Select first order. This means that the hypothesis node at time slice ( $t$ ) depends on the hypothesis node at time slice ( $t-1$ ); Figures D.6 & D.7.

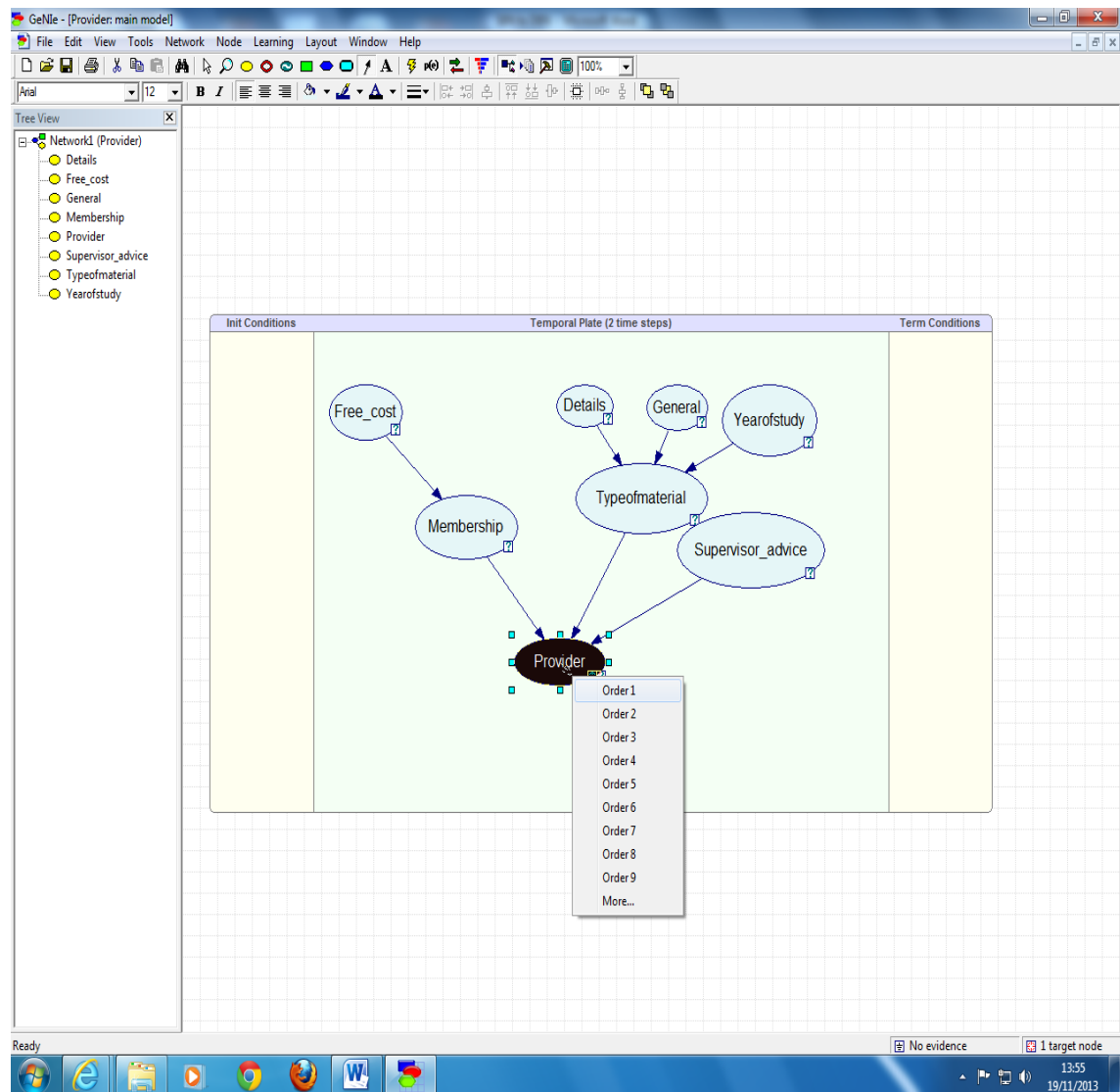


Figure D.6: Creating DBN, step 5 A

## Appendices

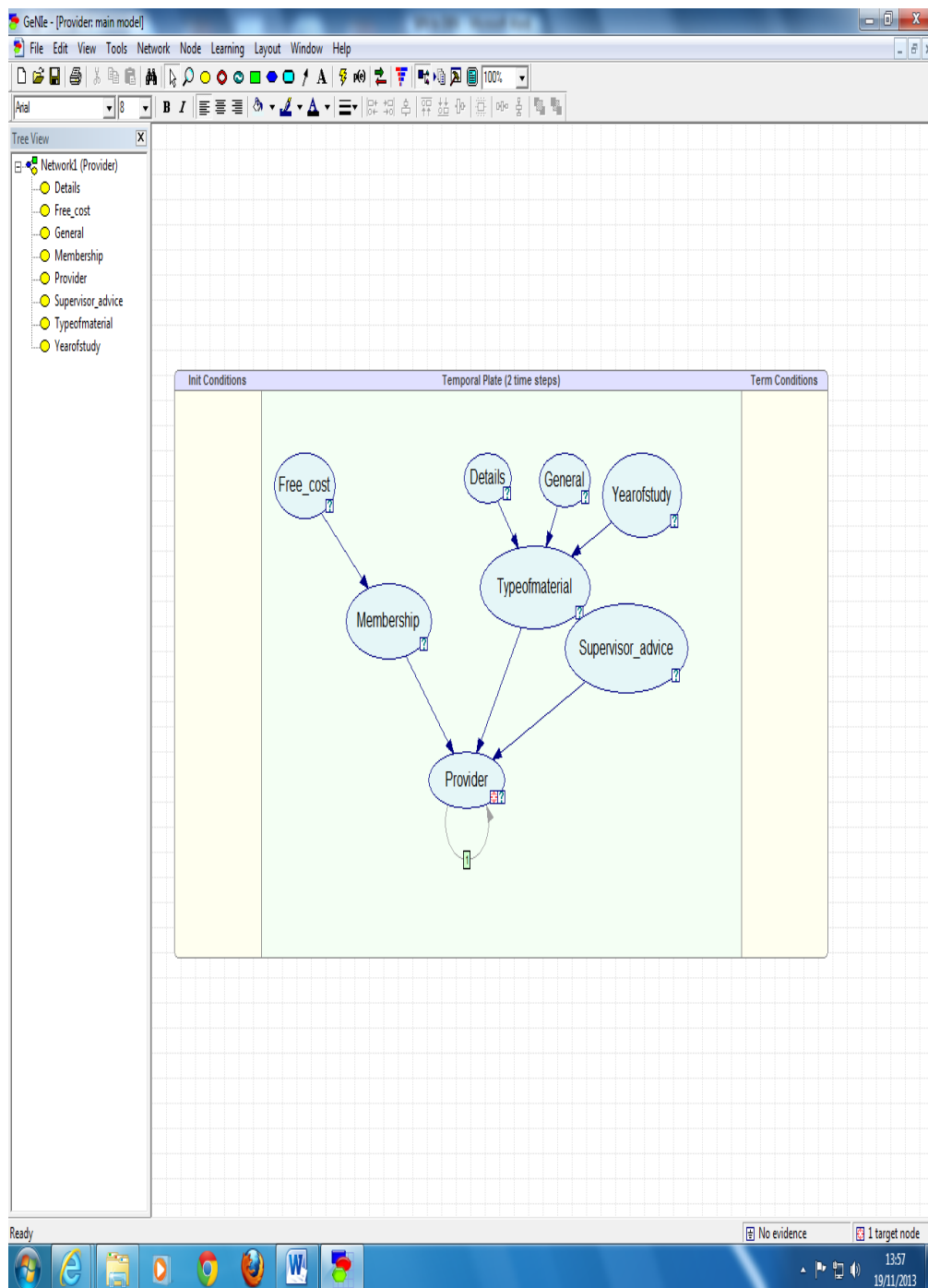


Figure D.7: Creating DBN, step 5 B

## Appendices

Step 6: Update inference (belief), as shown in Figure D.8.

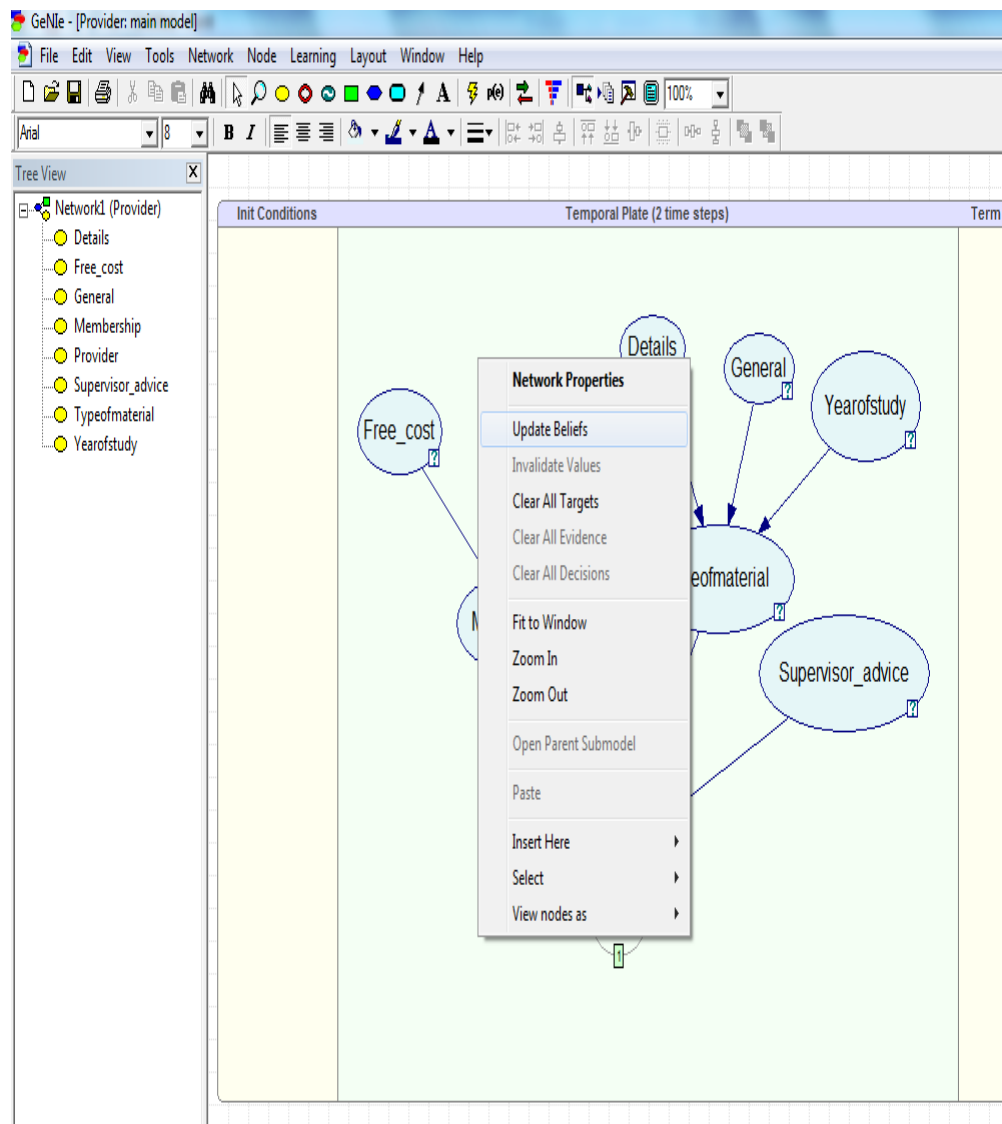


Figure D.8: Creating DBN, step 6